IRT Software: Überblick und Anwendungen

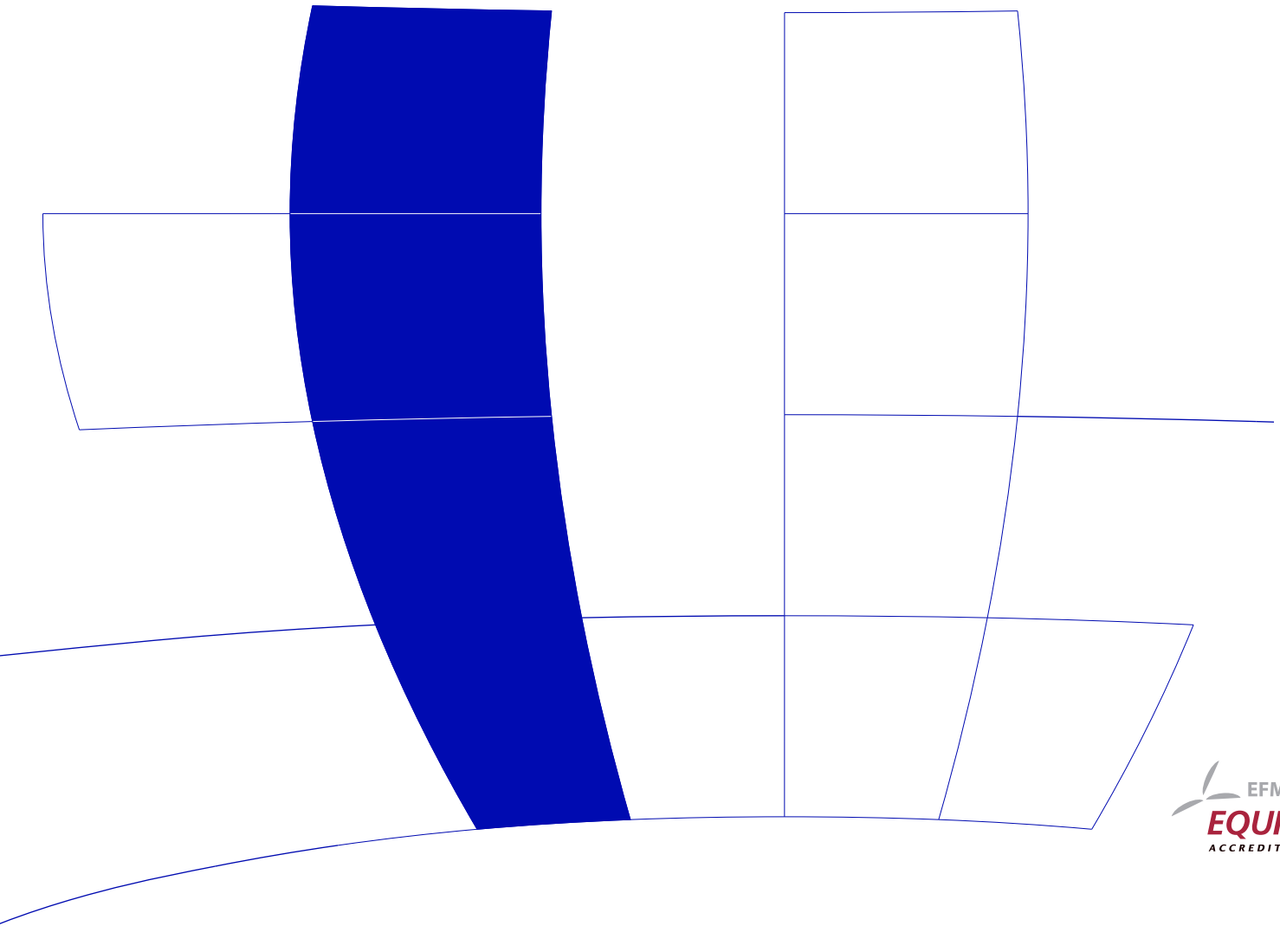
Marco Maier, Reinhold Hatzinger (Eds.)

Institute for Statistics and Mathematics
WU Wirtschaftsuniversität Wien

Research Report Series

Report 103
October 2010

http://statmath.wu.ac.at/



Vorwort

Diese Publikation wurde im Rahmen des Seminars *Psychometric Methods* erstellt. Dabei handelt es sich um eine Lehrveranstaltung, die jedes Semester am Institut für Statistik und Mathematik der Wirtschaftsuniversität Wien – mit wechselnden thematischen Schwerpunkten – abgehalten wird. Im Wintersemester 2009/2010 lag der Fokus auf der Anwendung von Item-Response-Software.

Zur Anwendung psychometrischer Methoden steht eine Vielzahl von Programmen zur Verfügung, die jeweils unterschiedliche Verfahren und Modelle anbieten. In diesem Seminar ging es im Wesentlichen darum, einen Überblick über die vorhandene Software zu bekommen, sowie die Stärken und Schwächen der einzelnen Programme herauszuarbeiten. Weiters sollten die Teilnehmer in die Lage versetzt werden, verschiedene psychometrische Modelle bei unterschiedlichen Problemstellungen praktisch anzuwenden.

Im Rahmen des Seminars wurden von verschiedenen Teilnehmergruppen jeweils ein bestimmtes Programm vorgestellt. Einerseits wurden die theoretischen Hintergründe und Modelle aufbereitetet und andererseits die jeweiligen Programme mittels Live-Präsentationen von Datenanalysen vorgeführt. Dadurch bekamen alle Beteiligten einen Einblick, welche Modelle in den unterschiedlichen Softwarepaketen umgesetzt sind, wie man sie anwenden und interpretieren kann und auch, wie man praktisch mit ihnen umgeht.

Damit die gewonnenen Erfahrungen auch für andere nutzbar werden haben wir die Gruppenbeiträge gesammelt herausgegeben. Die einzelnen Kapitel sollen jeweils eine Brücke zwischen den theoretisch- technischen Aspekten und anwendungsorientierten-praktischen Aspekten der einzelnen Progamme schlagen. Wichtig war uns auch die Auswahl der vorgestellten Softwarepakete, wobei sich der Bogen von etablierten und weitverbreiteten Programmen (z.B. BILOG oder MULTILOG) bis zu eher selten verwendenten Programmen (bspw. GGUM oder ScoRight) spannt.

Ohne Anspruch auf Vollständigkeit hoffen wir mit diesem Buch einen Einblick in die wichtigsten Softwarepakete zu geben, wobei wir auf eine verständliche Erklärung theoretischer Hintergründe und möglichst interessante Anwendungsbeispiele großen Wert legten. Unser Ziel war es, interessierten Anwenderinnen und Anwendern eine kleine ‚Landkarte' durch den Dschungel verfügbarer IRT-Software bereitzustellen, die zur weiteren Vertiefung anregen soll.

Unser Dank gilt den Teilnehmerinnen und Teilnehmern des Seminars, die ihre Beiträge mit viel Engagement und Durchhaltevermögen (für nicht wenige war dieser Artikel die erste Begegnung mit LaTeX) verfasst und überarbeitet haben, sodass dieses Werk zustande kommen konnte.

Wien, 2010 Marco Maier & Reinhold Hatzinger

Inhaltsverzeichnis

1 **Winsteps** 7
Stefan Angel und Richard Mühlmann
1.1 Winsteps 7
1.2 Rasch-Modell 8
1.3 Programmerläuterung 9
1.3.1 Beispieldaten 9
1.3.2 Dateninput und Modellspezifikation 11
1.3.3 Analyse – Interpretation der Outputs 14
1.3.4 Grafiken 17
1.4 Schlussbetrachtung 17

2 **Scoright** 27
Kathrin Gruber
2.1 Einleitung 27
2.2 Klassische Testlet Response Modelle 28
2.3 Bayes'sche Testlet Response Modelle 28
2.3.1 Der Testletparameter 29
2.3.2 Die Parameterschätzung 29
2.4 Das Programmpaket SCORIGHT 30
2.4.1 Die Verwendung von SCORIGHT 31
2.5 Diskussion 38

3 **FACETS Rasch Software illustriert am Beispiel von Multiple-Choice-Tests in Marketing** 39
Margit Kastner, Amata Ring und Brigitte Stangl
3.1 Einleitung 39
3.2 Rasch-Modell Grundlagen 40
3.3 Multifacetten-Rasch-Modell 42
3.4 Beschreibung von FACETS 43
3.5 Exemplarische Darstellung einer Anwendung 45
3.5.1 Daten 45
3.5.2 Modellspezifikation 46
3.5.3 Interpretation der Ergebnisse 50
3.6 Diskussion 55

4 **WinMiRa 2001** 59
Karl Ledermüller, Thomas Peschta und Wolfgang Ziniel
4.1 Theorie Raschmodelle 59
4.1.1 Messtheoretische Fundierung 59
4.1.2 Wichtige Vorteile Probabilistischer Testmodelle 60
4.1.3 Erweiterungen des klassischen Rasch Modells 60

4.1.4 Annahmen über threshold parameter im polytomen (Mixed) Rasch Modell . 61
4.2 Programmbeschreibung 63
4.2.1 unterstützte Modelle 63
4.2.2 unterstützte Skalenniveaus 64
4.2.3 Usability 64
4.2.4 grundsätzliche Bedienung 65
4.3 Beispiele 66
4.3.1 Latent Class Analyse mit dichotomen Daten 66
4.3.2 Mixed Rasch Modell mit WinMiRa 71
4.4 Conclusio 77

5 **BILOG-MG und MULTILOG** 79
Marco J. Maier und Konradin Maier
5.1 Einleitung 79
5.2 Modelle 79
5.2.1 BILOG-MG 79
5.2.2 MULTILOG 80
5.3 Parameterschätzung 82
5.3.1 Itemparameter 82
5.3.2 Personenparameter 83
5.4 Bedienung der Software und Anwendungsbeispiele 83
5.4.1 BILOG-MG 83
5.4.2 MULTILOG 89
5.5 Diskussion 93

6 **Extended Rasch Modeling: The R Package `eRm`** 95
Patrick Mair und Reinhold Hatzinger
6.1 Introduction 95
6.2 Extended Rasch models 96
6.2.1 General expressions 96
6.2.2 Representation of extended Rasch models 96
6.2.3 The concept of virtual items 98
6.3 Estimation of item and person parameters 100
6.3.1 CML for item parameter estimation 100
6.3.2 Mathematical properties of the CML estimates 101
6.3.3 CML and specific objectivity 103
6.3.4 Estimation of person parameters 104
6.4 Testing extended Rasch models 105
6.4.1 Item-fit and person-fit statistics 105
6.4.2 A Wald test for item elimination 105
6.4.3 Andersen's likelihood-ratio test 106
6.4.4 Nonparametric ("exact") Tests 106
6.4.5 Martin-Löf Test 106
6.5 The eRm package and application examples 106
6.5.1 Structure of the eRm package 106
6.5.2 Example 1: Rasch model 107
6.5.3 Example 2: LLTM as a restricted Rasch model 109
6.5.4 Example 3: RSM and PCM 109
6.5.5 An LPCM for repeated measurements in different groups 112
6.6 Additional topics 113
6.6.1 The eRm simulation module 113
6.7 Discussion and outlook 114

7 **ACER ConQuest – Generalised Item Response Modelling Software** 117
Bernadett Pauer, Julia Anette von Poswik und Thomas Rusch
7.1 ACER ConQuest - Eine Einführung 117
7.1.1 Modellierungen in ConQuest 117
7.2 Modell und Schätzung 118
7.2.1 Das multidimensionale multinominale Logit-Modell mit zufallsbedingten Koeffizienten 118
7.2.2 Das Populationsmodell 119
7.2.3 Schätzung 119
7.2.4 Der Aufbau von ACER Conquest 120
7.3 Anwendungen 120
7.3.1 Beispiel 1: Dichotomer Multiple Choice Test 120
7.3.2 Beispiel 2: Die Analyse von Beurteilungseffekten 125
7.3.3 Beispiel 3: Mehrdimensionale latente Regression 126

8 **IRT Analysis using the `ltm` Package** 131
Dimitris Rizopoulos
8.1 Basics of IRT 131
8.1.1 IRT Models for Dichotomous Data 131
8.1.2 IRT Models for Polytomous Data 132
8.1.3 Marginal Maximum Likelihood Estimation 133
8.1.4 Estimating the Latent Variable 133
8.2 IRT Analysis for Dichotomous Data 133
8.2.1 Descriptive Analysis 134
8.2.2 IRT Models for Dichotomous Data 136
8.3 IRT Analysis for Polytomous Data 148
8.3.1 Descriptive Analysis 148
8.3.2 IRT Models for Polytomous Data 149

9 **RUMM – Rasch Unidimensional Measurement Models** 159
Thomas Salzberger
9.1 Grundlagen 159
9.2 Modelle 159
9.3 Schätzmethode 160
9.4 Dateninput 160
9.5 Analysen 161
9.6 Ergebnisse 162
9.6.1 Item Parameter Details 162
9.6.2 Fittests 163
9.7 Grafische Ausgaben 166
9.8 Resumee 169

10 **GGUM2004 – Generalized Graded Unfolding Models** 171
Axel Sonntag und Daniela Weber
10.1 Die Methode der Enfaltungstechnik 171
10.1.1 Generalized Graded Unfolding Models 172
10.1.2 Parameter-Schätzung 173
10.2 Das Programm GGUM 2004 174
10.2.1 Spezifikation einer Schätzung 174
10.2.2 Ein Beispieldatensatz 178
10.2.3 Weitere (kommerzielle) Softwarepakete zur Schätzung von Unfolding Models 181
Literaturverzeichnis 183

Kapitel 1
Winsteps

Stefan Angel und Richard Mühlmann

Zusammenfassung Dieser Beitrag gibt einen kurzen Überblick über die Rasch-Analysesoftware "Winsteps". Die grundlegenden Funktionen des Programms werden anhand zweier Beispieldatensätze aus der Pflegeforschung illustriert. Dabei wird auf den Import von Daten aus anderen Dateiformaten, die Möglichkeiten der Modellspezifikation sowie auf eine Auswahl der verfügbaren Auswertungen eingegangen. Stärken der Software liegen in den vielfältigen implementierten Möglichkeiten zur Analyse der geschätzten Modelle sowie in der vergleichsweise einfachen Bedienung. Auch die Datenverarbeitungskapazität und der einfache Export der Ergebnisse sind als positiv zu werten. Als Schwächen des Programms können die eingeschränkte Zahl an spezifizierbaren Modellen, eine gewisse Redundanz und somit Unübersichtlichkeit in den Optionen zur Darstellung der Ergebnisse sowie eventuell – im Fall professioneller Anwender – die Beschränkung auf einen einzigen Schätzalgorithmus genannt werden.

1.1 Winsteps

Die Entwicklung der Vorläufer von WINSTEPS begann 1983 durch Benjamin Wright (den Pionier der Item Response Theory in den USA) sowie Mike Linacre im Rahmen eines Consultingauftrags. 1984 resultierte aus dieser Arbeit die erste Version einer Rasch-Analysesoftware, zu diesem Zeitpunkt noch als Add-on zu einer gängigen Tabellenkalkulationssoftware.

Deren Limitationen erzwangen die Weiterentwicklung zu einem eigenständigen DOS-Programm, Microscale. Dieses wurde 1987 zu einer Folgeversion für UNIX-basierte Rechner (MSCALE) umgeschrieben und aufgrund des Aufkommens des PCs sowie um einige Funktionen erweitert 1989 als BIGSCALE veröffentlicht. Dieses Programm wurde wiederum um einige Funktionen erweitert ab 1991 unter dem Namen BIGSTEPS vertrieben.

Die Adaptierung von BIGSTEPS für Windows führte schlussendlich 1998 zur Erstellung von "Winsteps". Abwärtskompatibilität ist bis zur Version MSTEPS gegeben (Linacre, 2004b).

Die aktuelle Version von Winsteps verfügt über beträchtliche Datenkapazitäten, es können bis zu 60.000 Items mit bis zu 255 Ausprägungen von bis zu 10.000.000 Personen analysiert werden.

Typischerweise mit Winsteps operationalisierte Rasch-Modelle sind das dichotome Rasch-Modell sowie polytome Rating Scale, Partial Credit oder Grouped Response-Structure Modelle.

Auch die Berechnung von komplexeren Modellen wie beispielsweise Paarvergleichsmodellen ist in Winsteps möglich, jedoch wird die Anwendung der speziell für komplexere Analysen erstellten Programmvariante "Facets" ausdrücklich empfohlen.

Die Verwendung des Programmes macht diesen Fokus auf Standardanalysen insofern ebenfalls ersichtlich, als die Modellspezifikation ausschließlich per Code möglich ist, während für auf die

WU Wien, Institut für Sozialpolitik • stefan.angel@wu.ac.at
WU Wien, Forschungsinstitut für Altersökonomie • richard.muehlmann@wu.ac.at

Schätzung folgende Auswertungen eine Vielzahl von Tabellen und Grafiken menübasiert aufgerufen werden kann.

Alternativ zur Vollversion des Programms steht eine Studierendenversion namens "Ministeps" zum freien Download zur Verfügung. Diese verfügt zwar über sämtliche Funktionen von Winsteps, die Datenkapazität ist mit 25 Items und 75 Personen jedoch stark reduziert. Somit kann diese Version zwar als hilfreich erachtet werden, um die Funktionalität von Winsteps zu erlernen, ihrer Anwendung zu analytischen Zwecken steht aber vor allem die geringe Zahl an Personen entgegen.

Darüeber hinaus ist eine Version von Winsteps verfügbar, die dem Buch "Applying The Rasch Model" von (Bond und Fox, 2007) beigefügt ist und als "Bond&FoxSteps" bezeichnet wird. Diese verfügt zwar über die vollständige Datenkapazität, jedoch – dem Anspruch des Buches als Einsteigerliteratur entsprechend – nicht über die volle Funktonalität von Winsteps.

Die vollwertige Version von Winsteps ist zum Preis von 149$ als Download von http://www.winsteps.com/ erhältlich.

1.2 Rasch-Modell

Mit Winsteps können folgende Modelle geschätzt werden: Dichotome Rasch-Modelle, Andrich Rating Scale-Modelle, Partial Credit Modell, Glas-Verhelst "Succes"-Modelle (auch "Steps"-Modelle genannt), Linacre "Failure" (mastery)-Modelle sowie Kombinationen dieser Modelltypen.

Im dichotomen Rasch-Modell wird die Wahrscheinlichkeit, dass eine Person v ein binäres Testitem i richtig beantwortet (Itemscore $x=1$) als Funktion der Itemschwierigkeit β_i und der Fähigkeit der Person θ_v angenommen, sodass gilt

$$\Pr(x = 1) = f(\theta_v - \beta_i) \tag{1.1}$$

Das dichotome Rasch-Modell gibt somit die bedingte Erfolgswahrscheinlichkeit einer Person v bei einem Item i wieder, gegeben deren Fähigkeitswert und gegeben die Itemschwierigkeit. Weiters geht das Rasch-Modell von einem logarithmischen Zusammenhang aus, sodass gilt

$$\Pr(x_{vi}|\theta_v, \beta_i) = \frac{e^{(\theta_v - \beta_i)}}{1 + e^{(\theta_v - \beta_i)}} \tag{1.2}$$

Rating Scale-Modelle stellen eine Erweiterung des dichotomen Rasch-Modells dar, für den Fall dass Items mehr als zwei Antwortkategorien aufweisen (z. B. Likert-Skalen). Für jedes Item mit z Kategorien werden $z - 1 = h$ Schwellenwerte modelliert, für die jeweils ein eigener Schwierigkeitsparameter β_{ih} geschätzt wird. Dieser Schätzer stellt den Schwellenwert dar, bei dem eine Person eine 50/50 Chance hat, sich für eine von zwei angrenzenden Kategorien zu entscheiden.

Für die Beispieldaten in diesem Beitrag soll ein (Masters') Partial-Credit-Modell geschätzt werden, das nicht nur polytome Items, sondern auch unterschiedliche Schrittweiten zwischen den Antwortkategorien der Items zulässt. Mit Hilfe dieses Modells kann nicht nur ermittelt werden, ob die abgefragten Kategorien das gleiche latente Konstrukt messen, sondern auch kontrolliert werden, ob die Schrittweiten passend gewählt wurden.

Das Partial-Credit-Modell hat die Form

$$\Pr(X_{vi} = h) = \frac{e^{(h\theta_h + \beta_{ih})}}{\sum_{l=0}^{m_i} e^{(l\theta_v + \beta_{il})}} \tag{1.3}$$

wobei $m_i = h - 1$ und $\beta_{ih} = -\sum_{j=0}^{h} \delta_{ij}$. Die Gleichung für die einzelnen Schwellenwerte lautet somit:

$$\Pr(x_{vij}|\theta_v, \delta_{ij}) = \frac{e^{(\theta_v - \delta_{ij})}}{1 + e^{(\theta_v - \delta_{ij})}} \tag{1.4}$$

Mair und Treiblmaier (2008) wobei jedem Item i ein Set von j Schwellenwerten zwischen Antwortkategorien zugeordnet wird.

Das Glas-Verhelst "Success" (growth)-Modell (auch "Steps"- Modell; Verhelst et al. (1997)), zielt auf hierarchische Skalen ab, bei denen Personen ein bestimmtes Item erst dann erreichen, wenn ein zuvor abgefragtes Item korrekt beantwortet wurde. Das Linacre "Failure" (mastery)-Modell (Linacre, 1991) eignet sich für die Anwendung auf Skalen, bei denen eine Person bei einem korrekt gelösten Item den Maximalwert zugewiesen bekommt und keine weiteren Items abgefragt werden. Bei einem falsch gelösten Item werden fortlaufend Items mit geringerem Schwierigkeitsgrad vorgelegt, bis eine korrekte Lösung erzielt werden kann.

1.3 Programmerläuterung

1.3.1 Beispieldaten

Die Software soll anhand zweier Beispiele illustriert werden. Die in diesen Beispielen analysierten Daten stammen aus der Wiener Studie zur informellen Pflege und Betreuung älterer Menschen 2008 (Trukeschitz et al., 2009), bei der Hauptbetreuungspersonen aus der Familie oder dem Freundes- und Bekanntenkreis befragt wurden. Die Beispieldatensets umfassen nur die Untergruppe jener, die die Pflegetätigkeit neben einer Erwerbstätigkeit ausüben, was sich in erster Linie durch einen höheren Grad an Vollständigkeit der Angaben dieser (relativ jungen und kognitiv leitungsfähigen) Gruppe begründet.

Das Rasch-Modell ist im Stande, Fälle auch bei fehlenden Werten in die Analyse mit einzubeziehen. Auch Winsteps verfügt über diese Fähigkeit. Jedoch wird in den hier durchgeführten Analysen auf unvollständige Fälle verzichtet, worin auch die Abweichung in der Fallzahl zwischen den beiden empirischen Beispielen begründet ist.

Das dichotome Rasch-Modell soll anhand von Daten zum Pflegebedarf der betreuten Personen illustriert werden. Dafür wird ein (I)ADL-Index ((Instrumental) Activities of Daily Living) (Katz et al., 1963) herangezogen. In der sozialwissenschaftlichen Literatur zur Pflege und Betreuung älterer Menschen werden mit Hilfe des (I)ADL-Index sowohl der Grad des Betreuungsbedarfs bzw. der Einschränkungen einer Person, wie auch das Ausmaß der für diese Person erbrachten Hilfestellungen dargestellt. Im gegebenen Fall soll untersucht werden, inwieweit der Index zur Messung des Betreuungsbedarfs herangezogen werden kann. Dabei wird der Pflegebedarf von betreuungsbedürftigen Personen anhand der Frage erhoben, ob die Fähigkeit vorliegt bzw. fehlt, gewisse Aktivitäten des täglichen Lebens ohne Hilfe durchzuführen.

Der Umstand, dass Hilfe bei einer der in Tabelle 1 angeführten Tätigkeiten erforderlich ist, wird mit "1" codiert, "0" bezeichnet, dass hinsichtlich der genannten Tätigkeit kein Hilfebedarf vorliegt.

Die vorliegenden Daten stammen von 568 erwerbstätigen WienerInnen. Diese geben an, welchen Einschränkungen die von ihnen neben der Erwerbstätigkeit betreuten Personen aus der Familie oder dem Freundes- und Bekanntenkreis unterliegen. Mit der gleichen Itembatterie wurde zudem erhoben, welche Hilfeleistungen die befragten sowie andere Betreuungspersonen erbringen. Somit wurden die Antworten der Befragten umcodiert, da die Antwortkategorien im Fragebogen "Hilfe durch mich", "Hilfe von anderen Angehörigen / Bekannten", äußerfamiliäre oder professionelle Hilfe" bzw. "Betreute Person benötigt keine Hilfe" lauteten.

Tabelle 1 gibt einen Überblick über die zur Erstellung des Index abgefragten Tätigkeiten.

Die abgefragten Tätigkeiten decken ein breites Spektrum an Hilfs- und Pflegeleistungen ab und erstrecken sich von stark körperbezogenen, intensiv pflegerischen Tätigkeiten bis zu Besorgungen und diverser Hausarbeit. Weiters sind Items enthalten, die mehr auf die Erfassung kognitiver als körperlicher Einschränkungen abzielen (Items 5, 11, 12, 16, 17).

Aus der Tabelle wird auch die oben angesprochene Doppelfunktion der verwendeten Fragenbatterie ersichtlich. Beispielsweise kann Item 14 für die Untersuchung der Aufteilung von Hilfestellungen auf den Familien- und Bekanntenkreis durchaus Relevanz besitzen, im Kontext des Hilfebedarfs fällt eine Interpretation jedoch schwer.

Tabelle 1.1 Dichotome Items – (I)ADL-Index

Item	Beschreibung: Person benötigt Hilfe bei / mit ...
1	An- und Auskleiden
2	Körperpflege (baden, duschen, ...)
3	Zubereitung von Mahlzeiten
4	Einnahme der Mahlzeiten
5	Hilfe mit Medikamenten (erklären, erinnern, einnehmen,...)
6	Verrichtung der Notdurft bzw. Inkontinenzbetreuung
7	Hausarbeit (Wohnung reinigen, Wäsche waschen,...)
8	Besorgung von Lebensmitteln / Medikamenten / usw.
9	Hilfe bei der Fortbewegung au§erhalb der Wohnung
10	Hilfe bei der Fortbewegung innerhalb der Wohnung
11	Hilfe bei Geldangelegenheiten bzw. Bankgeschäften
12	Durchsehen der Post
13	Hilfe Organisation von (Arzttermine, andere Betreuung, ...)
14	Gesprächen / Unterhaltung
15	Emotionale Zuwendung, Zuspruch
16	Sozialkontakte erhalten (Freunde / Familie treffen,...)
17	Beaufsichtigung der betreuten Person

Quelle: VIC 2008

Als zweites Datenbeispiel soll das Partial-Credit-Modell anhand einer Kurzversion des Zarit Burden Interviews mit 4 Fragen (Bédard et al., 2001) betrachtet werden. Der Index zielt auf die Messung der psychischen Belastung von Pflegepersonen ab und besteht aus vier fünfstufigen Items. Diese erfragen, wie häufig ein bestimmtes Belastungsgefühl auftritt (siehe Tabelle 2). Dabei bedeutet ein Wert von "0", dass das entsprechende Gefühl nie auftritt, während ein Wert von "4" ausdrückt, dass das jeweilige Belastungsgefühl nahezu immer zu bemerken ist.

Tabelle 1.2 Polytome Items Ð Zarit Burden Interview, Screening Version, Übersetzung aus dem Englischen

Wie häufig haben Sie das Gefühl, das Sie ...	nie	selten	manchmal	öfters	nahezu immer
...aufgrund der Zeit, die Sie mit Ihrer/Ihrem betreuungsbedürftigen Angehörigen verbringen, nicht genug Zeit für sich selbst haben?					
...gestresst sind, wenn Sie versuchen, die Betreuung/Pflege mit anderen Verpflichtungen (Arbeit/Familie) zu vereinbaren?					
...angespannt sind, wenn Sie Ihre/n Angehörige/n betreuen/pflegen?					
unsicher sind, wie es mit der betreuten Person weitergehen wird?					

Quelle: VIC 2008

Die Daten stammen aus derselben Erhebung wie die Daten zum Pflege- bzw. Betreuungsbedarf älterer Menschen. 709 vollständige Fälle stehen für die Analyse zur Verfügung.

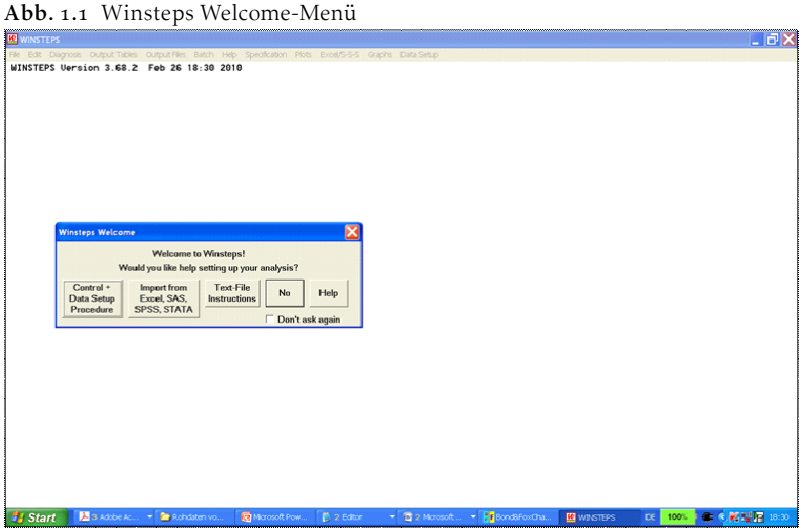
Eine Faktoranalyse (als in den Sozialwissenschaften etabliertes Alternativkonzept zum Rasch-Modell) gelangt zum Ergebnis, dass alle 5 Items einen Faktor laden, auch die Reliabilität ist mit einem Cronbach Alpha von 0,861 als hoch zu betrachten.

1.3.2 *Dateninput und Modellspezifikation*

Analysen in Winsteps benötigen als Input zwei verschiedene Komponenten: Daten und Prozessinformationen. Beide Informationen werden vom Anwender/von der Anwenderin gemeinsam in einem .txt-File spezifiziert, das von Winsteps zu einer Vielzahl an Outputs verarbeitet wird.

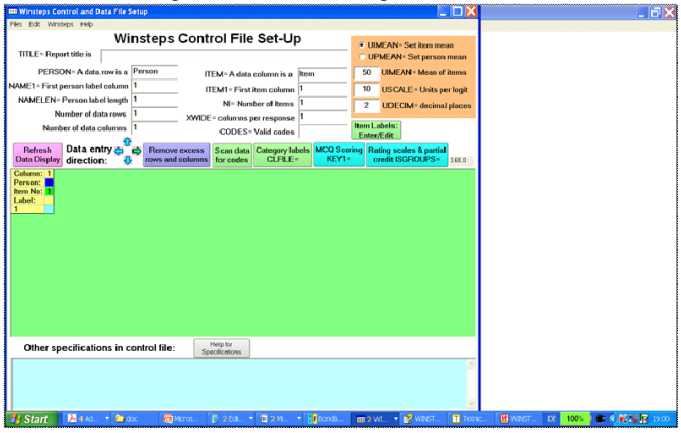
Nach dem Aufruf des Programmes gelangt man zunächst zum Hauptmenü (vgl. Abbildung 1). Wurde bereits ein Inputfile erzeugt, führt ein Klick auf "No" direkt zur Eingabeaufforderung der Inputdatei (vgl. Anmerkungen weiter unten).

Abb. 1.1 Winsteps Welcome-Menü



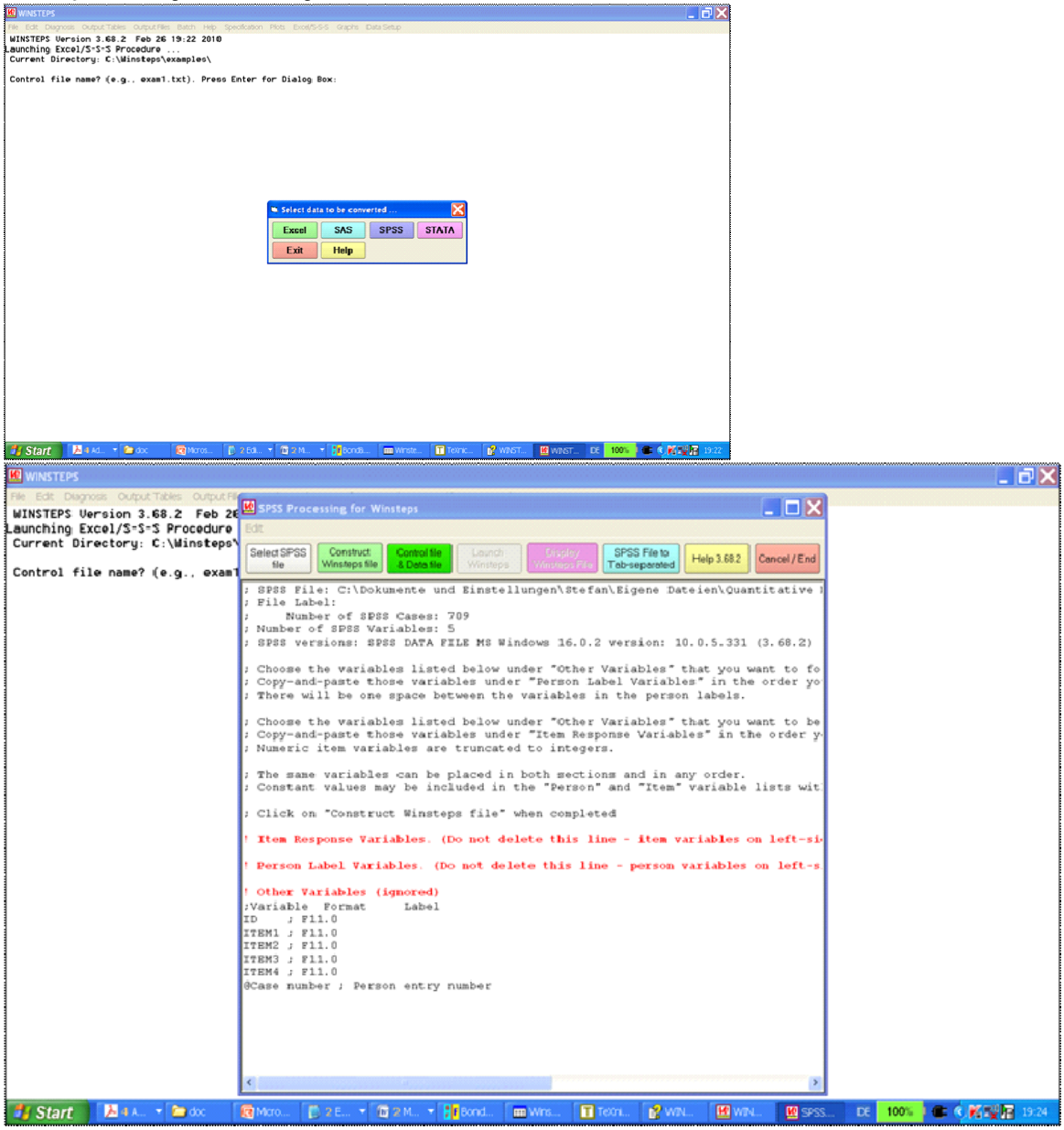
Nach der Auswahl von "Control + Data Setup Procedure" öffnet sich ein Interface, in dem mittels der Optionen von Winsteps ein Input-File neu generiert bzw. ein bereits vorhandenes Input-File um Daten oder Prozessinformation erweitert werden kann. Die verfügbaren Optionen beschränken sich aber nur auf einige zentrale Aspekte. Eine genaue Spezifikation des Inputfiles, insbesondere der Modellspezifikationen, ist erst nach Erzeugung im Editor selbst möglich (siehe Anmerkungen weiter unten). Sind alle Einstellungen abgeschlossen führt ein Klick auf die Optionen im Winsteps-Pull-Down-Menü zurück zum Winsteps-Hauptfenster (vgl. Abbildung 2).

Abb. 1.2 Winsteps Control File Set-Up


Winsteps Control and Data File Setup
Winsteps Control File Set-Up
TITLE= Report title is
PERSON= A data row is a
NAME1= First person label column
NAMELEN= Person label length
Number of data rows
Number of data columns
ITEM= A data column is a
ITEM1= First item column
NI= Number of items
XWIDE= columns per response
CODES= Valid codes
UIMEAN= Set item mean
UPMEAN= Set person mean
UMEAN= Mean of items
USCALE= Units per logit
UDECIM= decimal places
Item Labels: Enter/Edit
Refresh Data Display
Data entry direction:
Remove excess rows and columns
Scan data for codes
Category labels CLFILE=
MCQ Scoring KEY1=
Rating scales & partial credit ISGROUPS=
Other specifications in control file:
Help for Specifications

Die meisten User werden vermutlich ein bestehendes Datenfile in das .txt-Format umwandeln und um Prozessinformation ergänzen wollen. Winsteps kann importierte Daten in den Dateiformaten von Excel, SAS, SPSS, und STATA umwandeln und verarbeiten.

Abb. 1.3 Datenimport in Winsteps


WINSTEPS Version 3.68.2 Feb 26 19:22 2010
Launching Excel/S-S-S Procedure ...
Current Directory: C:\Winsteps\examples\
Control file name? (e.g., exam1.txt). Press Enter for Dialog Box:
Select data to be converted ...
Excel
SAS
SPSS
STATA
Exit
Help
WINSTEPS
SPSS Processing for Winsteps
Select SPSS file
Construct Winsteps file
Control file & Data file
Launch Winsteps
Display Winsteps File
SPSS File to Tab-separated
Help 3.68.2
Cancel / End
; SPSS File: C:\Dokumente und Einstellungen\Stefan\Eigene Dateien\Quantitative
; File Label:
; Number of SPSS Cases: 709
; Number of SPSS Variables: 5
; SPSS versions: SPSS DATA FILE MS Windows 16.0.2 version: 10.0.5.331 (3.68.2)
; Choose the variables listed below under "Other Variables" that you want to fo
; Copy-and-paste those variables under "Person Label Variables" in the order yo
; There will be one space between the variables in the person labels.
; Choose the variables listed below under "Other Variables" that you want to be
; Copy-and-paste those variables under "Item Response Variables" in the order y
; Numeric item variables are truncated to integers.
; The same variables can be placed in both sections and in any order.
; Constant values may be included in the "Person" and "Item" variable lists wit
; Click on "Construct Winsteps file" when completed
! Item Response Variables. (Do not delete this line - item variables on left-si
! Person Label Variables. (Do not delete this line - person variables on left-s
! Other Variables (ignored)
;Variable Format Label
ID ; F11.0
ITEM1 ; F11.0
ITEM2 ; F11.0
ITEM3 ; F11.0
ITEM4 ; F11.0
@Case number ; Person entry number

Winsteps erkennt die Anzahl an Items und die Identifikationsvariable automatisch. Die ID-Zeile muss unter die rote gefärbte Zeile "! Person Label Variables [. . .]" eingefügt (Cut and Paste) werden und die Itemliste unter "! Item Response Variables [. . .]". Durch einen Klick auf "Construct Winsteps file" wird auf Basis der hier verwendeten Daten ein Inputfile im .txt-Format erzeugt, das Daten und sog. Kontrollvariablen enthält (im Bereich "&Inst" bis "&End"), mittels derer die Analyse im Detail spezifiziert werden kann. Mittels " ; " können Kommentare in den Code eingefügt werden. Beim Import der Daten schlägt Winsteps automatisch einen bestimmten Modelltyp vor, der anschlie§end im Input-File geändert werden kann. Die genaue Form der Modellspezifikation kann über die Befehle "MODELS" und "GROUPS" bzw. "ISGROUPS" (synonym) eingegeben werden. "MODELS" stellt hierbei die erste Hierarchieebene bei den Einstellungen dar, die über "GROUPS"/"ISGROUPS" genauer kalibriert werden können. Bei der Kontrollvariable "MODELS" können die Optionen "R" (default), "S" und "F" eingetragen werden. "R" schätzt dichotome Rasch-Modelle, Andrich Rating Scale-Modelle und Masters' Partial-Credit-Modelle (Andrich, 1978b; Masters, 1982). "S" verwendet ein als Rasch-Modell (mit missing values bei den nicht erreichten Kategorien) parametrisiertes Glas-Verhelst SSuccess"(growth)-Modell (auch "Steps"- Modell; Verhelst et al. (1997)). Mit der Option

"F" schätzt Winsteps ein Linacre "Failure"(mastery)-Modell (Linacre, 1991). Das Linacre-Modell ist ebenfalls als Rasch-Modell parametrisiert mit fehlenden Werten bei den ungelösten Items.

Wenn die verfügbaren Codes bei "MODEL" nur einmal eingetragen werden, kommt das jeweilige Modell auf alle Items zur Anwendung. Alternativ können durch Eingabe von Strings für die einzelnen Items auch unterschiedliche Modelle geschätzt werden (z. B. bei vier Items: MODELS=RSSR; oder synonym MODELS=*1 R 2-3 S 4 R). Über die Kontrollvariable "GROUPS" / "ISGROUPS" kann dann innerhalb einer Modellgruppe (in dem hier verwendeten Beispiel "R") das exakte Schätzmodell bestimmt werden. Für den Zarit Burden Index wurde ein Partial Credit Modell (Code "0" für "GROUPS") und für die (I)ADL-Skala ein dichotomes Rasch-Modell geschätzt. Weitere verfügbare Optionen sind u.a. " " (Leerzeichen), wodurch ein Andrich Rating Scale-Modell berechnet wird.

Im Bereich zwischen "&END" und "END Names" kann jeweils pro Zeile ein Label für die verwendeten Items vergeben werden. Der Bereich nach "END NAMES" ist für die Daten reserviert, wobei eine Zeile einer Untersuchungseinheit entspricht und die Ausprägungen nacheinander ohne Abstand sowie die ID mit Abstand aufzulisten sind (erfolgt beim Datenimport automatisch).

Die aus dem Datenimportprozess für die hier verwendeten Daten resultierenden Inputfiles im .txt Format sind in den Abbildungen 1.4 sowie 1.5 zu sehen. Das von Winsteps generierte Textfile enthält allgemeine Informationen (Name des importierten Datenfiles, Zeitpunkt des Imports, Fallzahl, Zahl der Variablen), Spezifikationen zu den Daten (Spaltenbreiten in Zeichen, Festlegung der Label- und Datenspalten, die von den Daten angenommenen Werte) sowie zuletzt die importierten Daten. Wie in Abb. 1.5 zu sehen ist, wird auch der Modelltyp in den .txt-files festgelegt (GROUPS = 0).

Abb. 1.4 Controlfile (I)ADL

```
&INST
Title= ''D:\...\IADL.xls''
; Excel file created or last modified: tt.mm.yyyy hh:mm:ss
; IADL
;     Excel Cases processed = 568
; Excel Variables processed = 18
ITEM1 = 1 ; Starting column of item responses
NI = 17 ; Number of items
NAME1 = 19 ; Starting column for person label in data record
NAMLEN = 4 ; Length of person label
XWIDE = 1 ; Matches the widest data value observed
CODES = 01 ; matches the data
TOTALSCORE = Yes ; Include extreme responses in reported scores
; Person Label variables: columns in label: columns in line
@ID = 1E3 ; $C19W3
&END ; Item labels follow: columns in label
ADL1 ; Item 1 : 1-1
...
ADL17 ; Item 17 : 17-17
END NAMES
11111111111111111   1
01001011001111101   2
00000011101111111   3
[...]
```

Das Datenfile gibt den Pfad der eingelesenen (Excel-)Datei, die Zeit der Erstellung, die Fallzahl, die Zahl der eingelesenen Variablen, deren Position, Spaltenbreite und Spezifikation (Datenspalten, Namensspalte) sowie eine Beschreibung der Items an. Schlussendlich bilden die analysierten Daten das Ende des Files. Im Fall des (I)ADL-Index muss der Modelltyp nicht näher spezifiziert werden, da die dichotomen Daten ohnehin nur das Standardmodell zulassen.

Im Fall des Zarit Burden Index erstellt Winsteps die Kontrollvariable "GROUPS", da die Daten mehrere Modellspezifikationen zulassen. Winsteps schlägt hier das Partial-Credits-Modell vor (GROUPS = 0).

Abb. 1.5 Kontrollvarialben für Zarit Burden Index

```
&INST
Title= "C:\É\Zarit.sav"
; SPSS file created or last modified: tt.mm.yyyy hh:mm:ss
;
;    SPSS Cases processed = 709
; SPSS Variables processed = 5
ITEM1 = 1 ; Starting column of item responses
NI = 4 ; Number of items
NAME1 = 6 ; Starting column for person label in data record
NAMLEN = 4 ; Length of person label
XWIDE = 1 ; Matches the widest data value observed
GROUPS = 0 ; Partial Credit model: in case items have different rating scales
CODES = 01234 ; matches the data
TOTALSCORE = Yes ; Include extreme responses in reported scores
; Person Label variables: columns in label: columns in line
@ID = 1E3 ; $C6W3
&END ; Item labels follow: columns in label
ITEM1 ; Item 1 : 1-1
ITEM2 ; Item 2 : 2-2
ITEM3 ; Item 3 : 3-3
ITEM4 ; Item 4 : 4-4
END NAMES
4344   1
3324   2
3433   3
1220   4
2331   5
2212   6
3312   7
[...]
```

Nach dem Aufrufen der Input-Files über das Menü *File – Open File* könnten zunächst noch Dateinamen für die generierten Outputfiles vergeben werden, wobei hier das temporäre Speichern zu empfehlen ist (Bestätigung mit ENTER-Taste), da einzelne Outputs später ohnedies als .txt-Files gespeichert, nach der Berechnung über das *Output Files*-Menü spezifiziert, bzw. Grafiken in die Zwischenablage kopiert werden können. Auch bei der nachfolgenden Frage nach "Extra Specifications" empfehlen die Autoren des Programmes eine Bestätigung mittels ENTER-Taste. Danach wird die Berechnung gestartet, welche für die hier verwendeten Daten jeweils weniger als 1 Sekunde in Anspruch nimmt.

1.3.3 Analyse – Interpretation der Outputs

Die Benutzermenüs in Winsteps beziehen sich zum einen auf diverse Input- und Datenmodifkationen und zum anderen auf Outputs. An dieser Stelle sei angemerkt, dass in Winsteps mehrere Output-Informationen in den Menüs redundant vorkommen und der Schwerpunkt bei den nachfolgenden Anmerkungen zur Outputinterpretation deshalb überwiegend auf jenen Menüpunkten liegt, die (Bond und Fox, 2007) in der ihrem Rasch-Lehrbuch beiliegenden, von Winsteps abgeleiteten Software "Bond$FoxSteps" verfügbar machen.

Im *File*-Menü können Daten geladen, gespeichert und das Programm für eine zusätzliche Analyse neu geöffnet werden. Über das *Edit*-Menü ist ein schneller Zugriff auf das Inputfile möglich, falls Einstellungen bei den Kontrollvariablen manuell verändert werden sollen. Bei Verwendung des *Batch*-Menüs führt Winsteps Berechnungen im Batch-mode durch. Das Programm läuft dann im "Hintergrund" mit minimaler User-Interaktion. Im *Specification*-Menü können auch nach der Schätzung einige ausgewählte Spezifkationen des Schätzmodells mittels Kontrollvariablen-Code ab-

geändert werden. Unter *Output Files* kann der Winsteps-Output für die Weitervewendung in anderen Software-Paketen aufbereitet werden. Die Dialogfenster *Excel/S-S-S* bzw. *Data-Setup* führen zurück zum Datenimportfenster (vgl. Abbildung 3) bzw. zum Winsteps Control File Set-Up-Interface (vgl. Abbildung 2).

Für die Interpretation der Ergebnisse sind das *Diagnosis*-Menü, das *Output Tabels*-Menü sowie das *Graphs*-Menü von Bedeutung. Zentrale Informationen für die Interpretation der Analyseergebnisse finden sich im *Diagnosis*-Menü, auf das im Folgenden anhand der Beispieldaten näher eingegangen wird.

Abbildung 6 zeigt den Output zur Item-Polarität (*Diagnosis – A. Item Polarity*). Die Tabelle ist nach der Korrelation der Items mit dem gemessenen latenten Konstrukt sortiert (Spalte "PT-MEASURE"). Anhand der Korrelationskoeffizienten kann abgelesen werden, ob alle Items in der selben Richtung auf die latente Variable laden, was beim hier verwendeten Zarit Burden Index der Fall ist. Weites werden die Itemnummer, der Rawscore der Items, die Fallzahl (konstant, da keine Missings), das errechnete "Schwierigkeitsmaß" der Items, der Standardfehler, Infit Mean Square und Outfit Mean Square samt t-Statistiken ausgegeben. Dieselbe Tabelle findet sich – ergänzt um MOST MISFITTING RESPONSE STRINGS und MOST UNEXPECTED RESPONSES – im Menü *Diagnosis – E. Item Misfit Table*. Die gleiche Information kann graphisch auch unter *Diagnosis – B. Empirical Item-Category Measures* abgerufen werden.

Die zweite, in Abb. 6 abgekürzt dargestellte, Tabelle schlüsselt die Rawscores, durchschnittlichen Schwierigkeiten, durchschnittlichen Standardfehler, Outfit Measures sowie Punktkorrelationen nochmals nach den Antwortkategorien auf.

Abb. 1.6 Menü Diagnosis – *A. Item Polarity*

```
-------------------------------------------------------------------------------
PERSON: REAL SEP.: 1.88  REL.: .78 ... ITEM: REAL SEP.: 6.59  REL.: .98
         ITEM STATISTICS:  CORRELATION ORDER
-----------------------------------------------------------------------------------------------
|ENTRY   TOTAL                 MODEL|   INFIT  |  OUTFIT  |PT-MEASURE |EXACT MATCH|         |
|NUMBER  SCORE  COUNT  MEASURE  S.E. |MNSQ  ZSTD|MNSQ  ZSTD|CORR.  EXP.| OBS%  EXP%| ITEM  G |
|------------------------------------+----------+----------+-----------+-----------+---------|
|     4   1413    709    -.03     .06|1.54   8.6|1.58   8.9|  .75   .83| 40.3  51.9| ITEM4 0 |
|     3   1217    709     .63     .06| .82  -3.4| .83  -3.3|  .85   .83| 64.2  54.6| ITEM3 0 |
|     1   1428    709    -.08     .06| .85  -2.9| .85  -2.9|  .86   .83| 61.6  54.0| ITEM1 0 |
|     2   1564    709    -.51     .06| .75  -5.0| .74  -5.1|  .87   .83| 62.6  53.5| ITEM2 0 |
|------------------------------------+----------+----------+-----------+-----------+---------|
| MEAN  1405.5  709.0     .00     .06| .99   -.7|1.00   -.6|           | 57.2  53.5|         |
| S.D.   123.7    .0     .41     .00| .32   5.4| .34   5.5|           |  9.8   1.0|         |

         ITEM CATEGORY/OPTION/DISTRACTOR FREQUENCIES:  CORRELATION ORDER

-------------------------------------------------------------------
|ENTRY   DATA  SCORE |     DATA    | AVERAGE  S.E.  OUTF PTMEA|      |
|NUMBER  CODE  VALUE |  COUNT  %   | MEASURE  MEAN  MNSQ CORR.| ITEM |
|--------------------+-------------+--------------------------+------|
|    4   0         0 |    123  17  |   -3.05   .17  1.8  -.57 |ITEM4 |
|        1         1 |    122  17  |   -1.12   .13  1.8  -.19 |      |
|        2         2 |    191  27  |    -.27   .09  1.3  -.03 |      |
|        3         3 |    183  26  |    1.13   .11  1.6   .32 |      |
|        4         4 |     90  13  |    2.77   .19  1.4   .48 |      |
|                    |             |                          |      |
|    3   0         0 |    127  18  |   -3.40   .14   .8  -.65 |ITEM3 |
|        1         1 |    179  25  |   -1.02   .07   .8  -.22 |      |
|        2         2 |    214  30  |     .26   .07   .8   .12 |      |
|        3         3 |    146  21  |    1.93   .09   .8   .45 |      |
|        4         4 |     43   6  |    3.94   .25  1.1   .44 |      |
[...]
```

Im Menüpunkt *C. Category Function* wird für jedes Item der Skala geprüft, ob die durchschnittliche Fähigkeit der Personen, die in eine bestimmte Kategorie fallen, mit steigender Kategorie ebenfalls ansteigt. Falls dies nicht zutrifft, liegt die Vermutung nahe, dass die Kategorien einer Itemvariable nicht perfekt ordinal, d.h. dass bei steigenden Fähigkeitswerten nicht immer aufeinanderfolgene Kategorien mit der höchsten Wahrscheinlichkeit gewählt werden. Weiters finden sich in diesem Menüpunkt auch Item Characteristic Curves im ASCII-Format. Abbildung 7 veranschaulicht beispielhaft für Item 1 des Zarit Burden Indexes, dass Überschneidungsfreiheit gegeben ist (Werte in den Spalten "obsvd avrge" bzw. "score to measure/at Ca.").

Ein inhaltlich ähnlicher Output, der Informationen zur Hierarchie der Items enthält, lässt sich unter *F. Construct KeyMap* abrufen. Abbildung 8 veranschaulicht, wie hoch das durchschnittliche Rating für Personen liegt, welche eine bestimmte Ausprägung auf der latenten Variable aufweisen. Die unten in der Abbildung zu findenden Großbuchstaben "M", "S" und "T" zeigen die Lage der jeweiligen Mittelwerte (Means der Item- bzw. und Personenwerte) und die Abstände von einer ("S") bzw. zwei ("T") Standardabweichungen an. Auch Perzentile der Verteilung der Fähigkeitswerte werden dargestellt.

Unter *D. Dimensionality* werden 25 Tabellen ausgegeben, welche sich größtenteils wiederholende Informationen zur Frage beinhalten, ob alle Items die selbe (einzige) latente Dimension aufweisen. Winsteps identifiziert hier mögliche zusätzliche Dimensionen mittels einer Hauptkomponentenanalyse der beobachten Residuen. Die in Winsteps ermittelten Komponenten zeigen keine Ladungen auf Faktoren an, sondern Kontraste zwischen positiven und negativen Ladungen. Eine Komponente beinhaltet gegensätzliche Antwortmuster der Personen über die Items. Folglich kann diese Hauptkomponentenanalyse nicht in der selben Weise wie die klassische Faktorenanalyse interpretiert werden. Die Autoren von Winsteps weisen darauf hin, dass bis dato keine Kriterien entwickelt wurden, ab wann eine Abweichung das Vorliegen einer zusätzlichen Dimension anzeigt. Im Anwendungsbeispiel werden 63,5 % der Varianz durch das Rasch-Modell (erste Dimension) erklärt (vgl. Abbildung 9). Die größte zweite Dimension erklärt 16,8 % der Varianz ("unexplained variance in 1st Kontrast"). Linacre bieten als Orientierungshilfen für die Interpretation der Kennzahlen aus der Dimensionalitätsanalyse einige Richtwerte an: Eine erklärte Varianz durch den Rasch-Faktor von mehr als 50 % wird als "gut" und eine "unexplained variance explained by 1st Kontrast" < 5 % als "exzellent" eingestuft. Eigenwerte bei der "unexplained variance explained by 1st contrast" 3,0 bzw. 1,5 gelten als "gut" bzw. "exzellent". In der Analyse der (I)ADL Daten findet sich eine zusätzliche Dimension mit einem Eigenwert von 2,2. Diese erklärt jedoch nur 5,9% der Varianz (auf die Darstellung des Scree Plots wird aus Platzgründen verzichtet).

Die unter *Diagnosis – E. Item Misfit Table* erstellte Tabelle entspricht inhaltlich der Tabelle zur Itempolarität (siehe oben), unterscheidet sich aber in der Reihenfolge der Items, welche nach den Fehlerwerten sortiert ausgegeben werden.

Die im Menüpunkt *G. Person Misfit* enthaltene Auflistung einzelner Personen kann vor allem dazu dienen, anhand der Infit und Outfit-Kriterien einzelne Ausreißer in der Stichprobe zu identifizieren und aus der Berechnung auszuschließen. Weitere Indikatoren für Ausreißer finden sich in den hier ausgewiesenen Tabellen zu den "most unexpected observationsnd den "most misfitting response strings".

Diagnosis – H. Separation ermittelt, ob die Items die Fähigkeitswerte der Personen "treffen", d.h. Schwierigkeitsgrade aufweisen, die weder zu schweren noch zu einfachen "Aufgaben" entsprechen. Dafür werden sowohl Item- als auch Personenwerte mit und ohne extreme Antworten errechnet. Der Hintergrund für diese Differenzierung ist darin zu finden, dass Items, die von sämtlichen Personen gleich beantwortet werden, nicht zwischen Personen unterscheiden können. Analog kann auch kein Fähigkeitswert für Personen ermittelt werden, die sämtliche Items identisch beantworten.

Die gleiche Information ist auch unter *Output Tables – 12. Item: map* abrufbar. Dabei sind die Schwierigkeitswerte der Items auf der rechten und die Fähigkeitswerte der Personen auf der linken Seite zu sehen. Ein "#" repräsentiert dabei mehrere Personen.

Bei der Betrachtung des (I)ADL-Index wird ersichtlich, dass alle Personen zumindest eine Einschränkung ((I)ADL=1) haben. Jedoch repräsentiert beispielsweise Item 7 einen zu geringen Schwie-

rigkeitsgrad (im konkreten Fall: Grad der Einschränkung), um zwischen Personen zu diskriminieren. Andererseits hat eine bedeutende Zahl an Personen im untersuchten Datensatz zu starke Einschränkungen, um aufgrund der abgefragten Items unterschieden zu werden.

Die obige Darstellung erfolgt in umgekehrter Weise unter dem Menüpunkt *Output Tables – 16.PERSON: map*, wo die Fähigkeitswerte der Personen (ausgeschriebene IDs) gruppenweise den Werten der Items (dargestellt als "x") gegenüber gestellt werden. Aufgrund der großen Fallzahl der Anwendungsbeispiele geht die Darstellung in letzterem Fall jedoch der Übersichtlichkeit verlustig.

Unter *Output Tables – 17. PERSON: Measure* kann eine Reihung der Personen nach dem ebenfalls angezeigten Fähigkeitsparameter abgerufen werden. Wiederum bestehen Redundanzen in den Outputtabellen. So findet sich unter *Output Tables – 18. PERSON: entry* die selbe Liste, jedoch nach der ID sortiert.

1.3.4 Grafiken

Winsteps bietet im Menü *Graphs* mehrere Möglichkeiten zur graphischen Visualisierung der Schätzergebnisse. Unabhängig davon, welche der ersten neun Optionen des Drop-Down-Menüs ausgewählt wird, gelangt der Anwender/die Anwenderin zu einem grafischen Interface in dem sich neben der Grafik selbst auch alle Steuerelemente zur Auswahl unterschiedlicher Diagramme befinden. Es werden allerdings keine Diagrammüberschriften automatisch erzeugt.

Die Item Characteristic Curves (ICCs) können in Winsteps sowohl ohne als auch gemeinsam mit der empirischen ICC ausgegeben werden (inkl. Eines 95%igen Konfidenzinvervalles, vgl. Abbildung 11). Grundsätzlich werden alle Graphen für jedes Item einzeln dargestellt. Mit der Schaltfläche "Multiple Item ICCs" (vgl. Abbildung 11) ist es möglich, mehrere ICCs in einen einzigen Graphen zu plotten.

Eine vor allem für polytome Items wichtige Darstellungsform stellen die Probability Category Curves der Items dar. Diese können auch gemeinsam mit den empirischen Daten in jeder verwendeten Itemkategorie ausgegeben werden (vgl. Abbildung 12).

Die oberen Schnittpunkte der Graphen in Abb. 11 stellen jene Grenzwerte des Fähigkeitsparameters dar, ab denen die Wahrscheinlichkeit bei einem bestimmten Item die nächsthöhere Kategorie angekreuzt zu haben höher liegt als für die aktuelle Itemkategorie. Der Verlauf der Kurven bzw. die Lokalisierung ihrer Schnittpunkte ist somit auch ein Hinweis für die hierarchische Abfolge der jeweiligen Kategorien. Im Falle des Zarit Burden Indexes ergibt sich eine klare aufsteigende Hierarchie der Itemkategorien (was auch schon in den tabellarischen Outputs ersichtlich war, siehe oben). So liegt zum Beispiel der Schwellenwert des Fähigkeitsparameters (relativ zur Itemschwierigkeit) zwischen der dritten ("manchmal") und der vierten Kategorie ("öfters") im ersten Item bei ca. 0,7.

Die Auswahl der Schaltfläche "Cumulative Probabilities" (vgl. Abb. 13) zeigt für die Personen im Sample die kumulierte Wahrscheinlichkeit, in bestimmte Itemkategorien oder in eine der darunter liegenden Itemkategorien zu fallen, je nach Ausprägung des Fähigkeitsparameters. Die niedrigste Kategorie steht links. Für das Anwendungsbeispiel des Zarit Burden-Indexes bedeutet dies, dass Personen mit einem Fähigkeitsparameter von 2 eine 20%-ige Wahrscheinlichkeit haben, die dritte oder eine niedrigere Kategorie beim Item1 anzukreuzen. Etwas weniger als 20 von 100 RespondentInnen mit einem Fähigkeitsparameter von 5 kreuzen bei Item 1 die Kategorien 0, 1, 2, 3 oder 4 an.

1.4 Schlussbetrachtung

Das Ziel dieses Beitrages war es, auf zentrale Merkmale der für die Schätzung von Rasch-Modellen verfügbaren Software Winsteps einzugehen und anhand eines Beispiels die wichtigsten Analyseschritte im Programm für EinsteigerInnen bzw. potentielle UserInnen zu dokumentieren. In diesem abschließenden Kapitel sollen die Vorteile und Nachteile von Winsteps hinsichtlich der angebotenen Tools und ihrer Anwendung zusammenfaßsend gegenübergestellt werden.

Winsteps verzeichnet grundsätzlich eine relative hohe Datenverarbeitungskapazität. Die Berechnungsdauer für die hier verwendete vergleichsweise geringe Datenmenge lag bei weniger als einer Sekunde. Bei größeren Datensätzen kann sich der verfügbare Batch-Modus des Programms als günstig erweisen. Mittels Winsteps können sowohl dichotome als auch polytome Variablen analysiert werden. Die Schätzmöglichkeiten der Software sind relativ breit und reichen vom klassischen dichotomen Rasch-Modell über Rating Scale- und Partial-Credit-Modelle bis hin zu Glas-Verhelst-(Steps)Modellen und Linacre Failure-Modellen für hierarchische Skalen. Der verwendete JMLE-Schätzalogrithums kann allerdings nicht geändert werden.

Winsteps bietet relativ umfangreiche Möglichkeiten der grafischen Darstellung für die Schätzergebnisse. Sowohl die grafischen als auch die tabellarischen Outputs laßen sich problemfrei exportieren, was bei letzteren vor allem durch das verwendete .txt-Format begünstigt wird. Die grafischen User-Interfaces sind hilfreich bei der Editierung der Daten. Für die Modellspezifikation müssen alle Einstellungen jedoch überwiegend in den .txt-Files (Input-Files) vorgenommen werden. Manch User wird möglicherweise die Cut and Paste-Vorgehensweise beim Datenimport aus anderen Formaten als umständlich empfinden, insbesondere bei großen Datensätzen. Bezogen auf die über die Menüleiste abrufbaren Outputs lässt sich allerdings eine hohe Redundanz der ausgewiesenen Schätzergebnisse feststellen. Die große, sich oftmals überschneidende Zahl an .txt-Outputs, sowie eine nicht durchgehende Nummerierung der Menüoptionen bei den Output-Tabellen sind vor dem Hintergrund einer sehr großen Zahl an verfügbaren Output-Tabellen der Übersichtlichkeit nicht allzu förderlich.

Insgesamt verfügt Winsteps jedoch über ein sehr umfangreiches und detailliertes Hilfe-Dokument, in dem sich an den entsprechenden Stellen (v.a. bei den Schätzverfahren) auch Verweise auf wißenschaftliche Primärliteratur befinden. Darüber hinaus erleichtert die Verknüpfung einer anwendungsorientierten Einführung in die Rasch-Modellierung mit der Einführung in Winsteps im Buch von Bond und Fox (2007) den Einstieg in die Software. Angesichts der weiter oben erwähnten Anschaffungskosten ist die kostenlose, reduzierte Programmversion Ministeps bzw. das mit dem Erwerb des Buches von Bond und Fox (2007) verfügbare Programm Bond&FoxSteps möglicherweise ein nützliches Werkzeug zum Test der Programmfunktionen vor der Anschaffung der Vollversion.

Abb. 1.7 Output zu *C. Category Function*

```
SUMMARY OF CATEGORY STRUCTURE.  Model="R"
FOR GROUPING ''0'' ITEM NUMBER: 1  ITEM1
ITEM DIFFICULTY MEASURE OF -.08 ADDED TO MEASURES
-------------------------------------------------------------------
|CATEGORY   OBSERVED|OBSVD SAMPLE|INFIT OUTFIT||STRUCTURE|CATEGORY|
|LABEL SCORE COUNT %|AVRGE EXPECT|  MNSQ  MNSQ||CALIBRATN| MEASURE|
|-------------------+------------+------------++---------+--------|
|  0   0      54   8| -2.77 -2.60|   .84   .86||  NONE   |( -4.16)| 0
|  1   1     138  21| -1.46 -1.42|   .83   .83||   -2.86 |  -2.14 | 1
|  2   2     205  31|  -.22  -.19|   .78   .75||   -1.12 |   -.30 | 2
|  3   3     200  31|  1.36  1.25|   .84   .83||     .62 |   1.95 | 3
|  4   4      56   9|  2.76  2.76|  1.04  1.03||    3.37 |(  4.43)| 4
-------------------------------------------------------------------
OBSERVED AVERAGE is mean of measures in category. It is not a parameter estimate.

-------------------------------------------------------------------------------
|CATEGORY    STRUCTURE   |  SCORE-TO-MEASURE   | 50% CUM.| COHERENCE|ESTIM|
| LABEL    MEASURE  S.E. | AT CAT. ----ZONE----|PROBABLTY| M->C C->M|DISCR|
|------------------------+---------------------+---------+----------+-----|
|   0      NONE          |( -4.16) -INF   -3.28|         |  85%  31%|     | 0
|   1       -2.95    .17 |  -2.14  -3.28  -1.21|   -3.09 |  57%  55%| 1.18| 1
|   2       -1.21    .12 |   -.30  -1.21    .71|   -1.21 |  58%  68%| 1.21| 2
|   3         .53    .11 |   1.95    .71   3.44|     .62 |  63%  77%| 1.24| 3
|   4        3.28    .17 |(  4.43)  3.44  +INF |    3.34 |  86%  23%| 1.01| 4
-------------------------------------------------------------------------------
M->C = Does Measure imply Category?
C->M = Does Category imply Measure?

         CATEGORY PROBABILITIES: MODES - Structure measures at intersections
P      -+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-
R  1.0 +                                                             +
O      |                                                             |
B      |0                                                            |
A      | 000                                                        4|
B   .8 +    00                                                    44 +
I      |      0                                                  4   |
L      |       0                                               44    |
I      |        00                            3333333        4      |
T   .6 +          0                         3       33     4       +
Y      |           0    1111        222     33          33  4        |
    .5 +            0 11    11   222   22  3              34         +
O      |            1*        112        2*              443         |
F   .4 +           1  0       221        3 2            4   33       +
       |         11    0     2   1     33   22         4      3      |
R      |        1       0   2     11  3       2       4        3     |
E      |      11         002        13         22   44          33   |
S   .2 +    11           220        31           2 4              33 +
P      |  11           22   00    33  11         4*2               3|
O      |11           22       0033      11    444   222             |
N      |         2222       33330000      ****1        22222         |
S   .0 +********************44444444******00000**********************+
E      -+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-
       -5    -4    -3    -2    -1     0     1     2     3     4     5
        PERSON [MINUS] ITEM MEASURE

[...]
```

Abb. 1.8 F. Construct KeyMap

```
-------------------------------------------------------------------------------

EXPECTED SCORE: MEAN  (Rasch-score-point threshold, ":" indicates Rasch-half-point
threshold) (ILLUSTRATED BY AN OBSERVED CATEGORY)
-5       -3        -1         1         3         5         7
|---------+---------+---------+---------+---------+---------|  NUM   ITEM
0      0   :     1    :    2     :     3      :    4        4    3  ITEM3
|                                                           |
|                                                           |
0      0   :    1   :   2   :     3     :    4             4    4  ITEM4
0   0    :    1    :    2    :     3      :    4           4    1  ITEM1
|                                                           |
0   0   :    1   :   2    :     3      :   4               4    2  ITEM2
|---------+---------+---------+---------+---------+---------|  NUM   ITEM
-5       -3        -1         1         3         5         7

4    2    2  2 4 4 5  5 6 6  7  4  6  4   2    1      1
2    0    2  7 4 1 6  9 3 2  1  2  4  2   5    5      4        PERSONS
 T           S          M           S           T
0        10   20  30 40 50 60 70  80 90              99        PERCENTILE
[...]
```

Abb. 1.9 Überprüfung der Dimensionalität in Winsteps

```
-------------------------------------------------------------------------------

     Table of STANDARDIZED RESIDUAL variance (in Eigenvalue units)
                                            -- Empirical --    Modeled
Total raw variance in observations     =         11.0 100.0%         100.0%
  Raw variance explained by measures   =          7.0  63.5%          63.5%
    Raw variance explained by persons  =          4.8  43.6%          43.7%
    Raw Variance explained by items    =          2.2  19.8%          19.8%
  Raw unexplained variance (total)     =          4.0  36.5% 100.0%   36.5%
    Unexplned variance in 1st contrast =          1.8  16.8%  45.9%
    Unexplned variance in 2nd contrast =          1.3  11.6%  31.9%
    Unexplned variance in 3rd contrast =           .9   8.1%  22.0%
    Unexplned variance in 4th contrast =           .0    .0%    .1%
    Unexplned variance in 5th contrast =           .0    .0%    .0%

       STANDARDIZED RESIDUAL VARIANCE SCREE PLOT

        VARIANCE COMPONENT SCREE PLOT
       +--+--+--+--+--+--+--+--+--+--+
   100%+  T                           +
       |                              |
  V 63%+     M                        +
  A    |                              |
  R 40%+        P                     +
  I    |              U                |
  A 25%+                              +
  N    |                              |
  C 16%+           I     1            +
  E    |                              |
    10%+                    2         +
  L    |                        3     |
  O  6%+                              +
  G    |                              |
  |  4%+                              +
  S    |                              |
  C  3%+                              +
  A    |                              |
  L  2%+                              +
  E    |                              |
  D  1%+                              +
       |                              |
   0.5%+                         4  5  +
       +--+--+--+--+--+--+--+--+--+--+
          TV MV PV IV UV U1 U2 U3 U4 U5
         VARIANCE COMPONENTS
[...]
-------------------------------------------------------------------------------

     Table of STANDARDIZED RESIDUAL variance (in Eigenvalue units)
                                            -- Empirical --    Modeled
Total raw variance in observations     =         37.4 100.0%         100.0%
  Raw variance explained by measures   =         20.4  54.5%          45.0%
    Raw variance explained by persons  =          7.9  21.0%          17.4%
    Raw Variance explained by items    =         12.5  33.5%          27.6%
  Raw unexplained variance (total)     =         17.0  45.5% 100.0%   55.0%
    Unexplned variance in 1st contrast =          2.2   5.9%  12.9%
    Unexplned variance in 2nd contrast =          1.5   3.9%   8.7%
    Unexplned variance in 3rd contrast =          1.3   3.4%   7.6%
    Unexplned variance in 4th contrast =          1.2   3.1%   6.9%
    Unexplned variance in 5th contrast =          1.1   3.0%   6.7%
[...]
```

Abb. 1.10 Item Map (I)ADL-Index und Zarit Burden Index

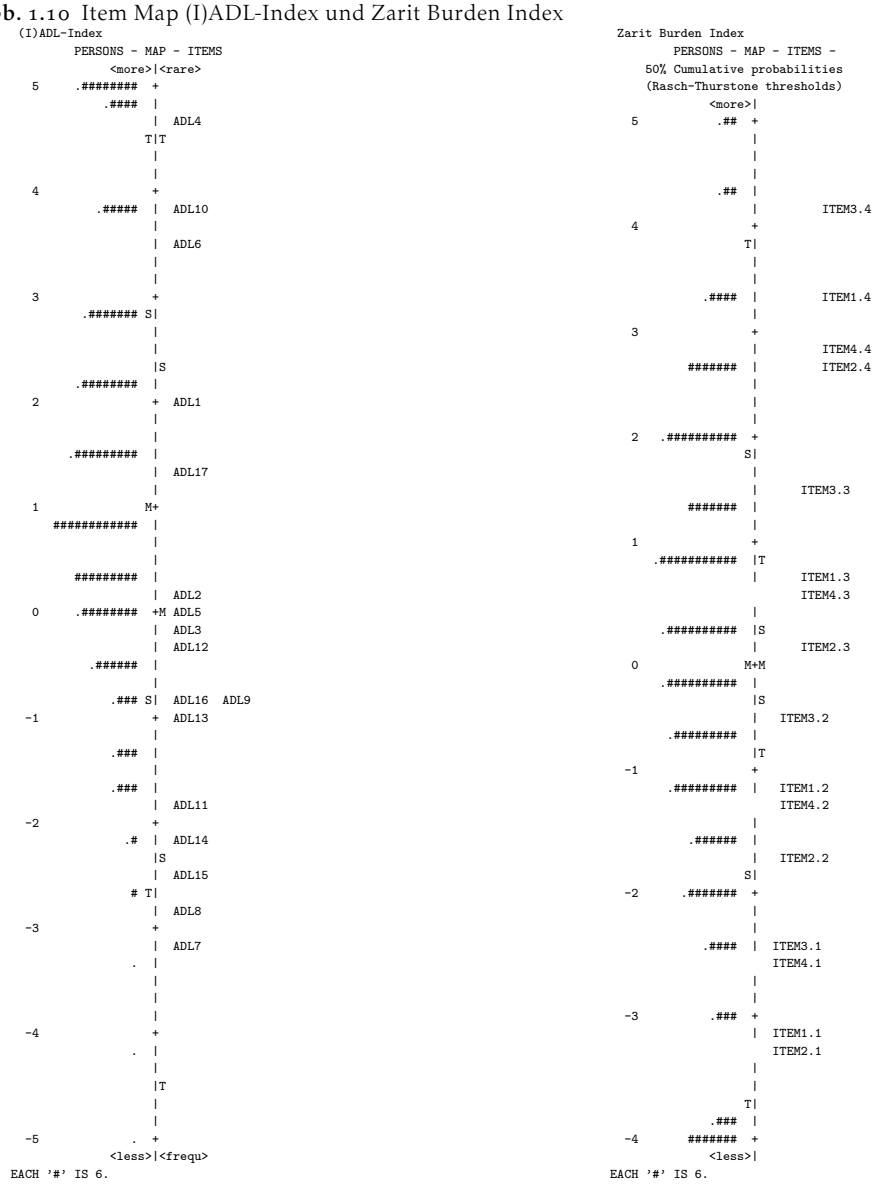


Abb. 1.11 Das Graphen-Fenster in Winsteps

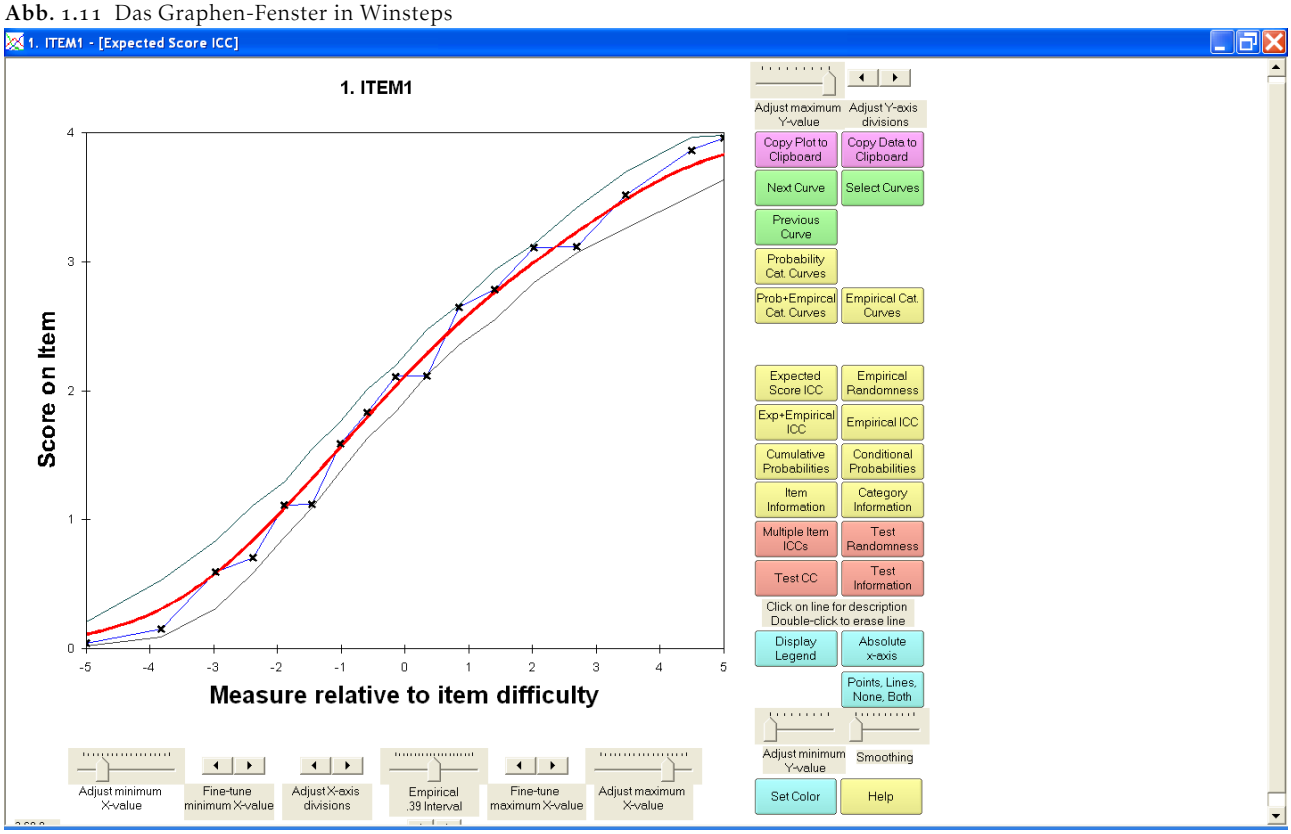


Abb. 1.12 Probability and Empirical Category Curve

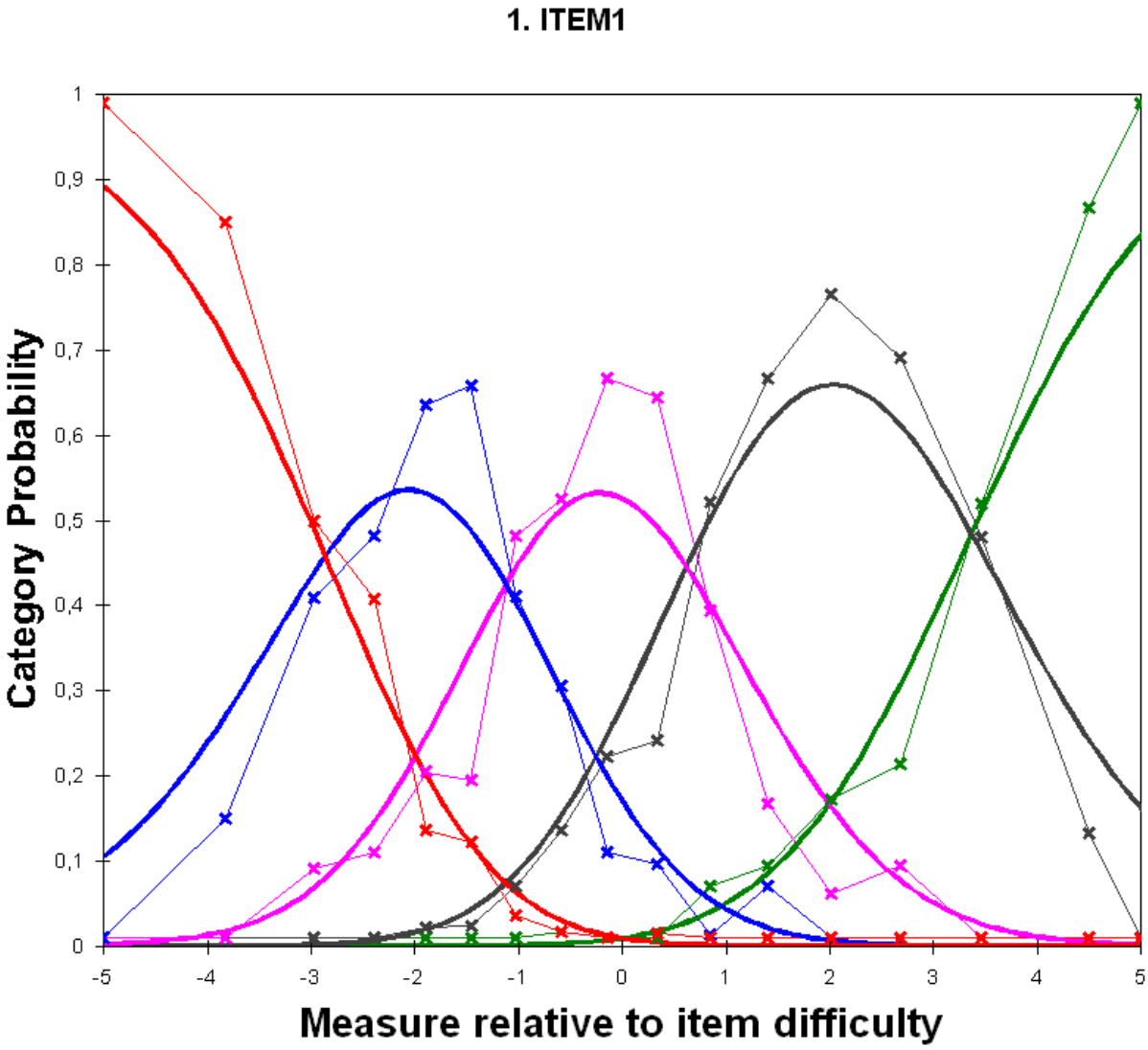
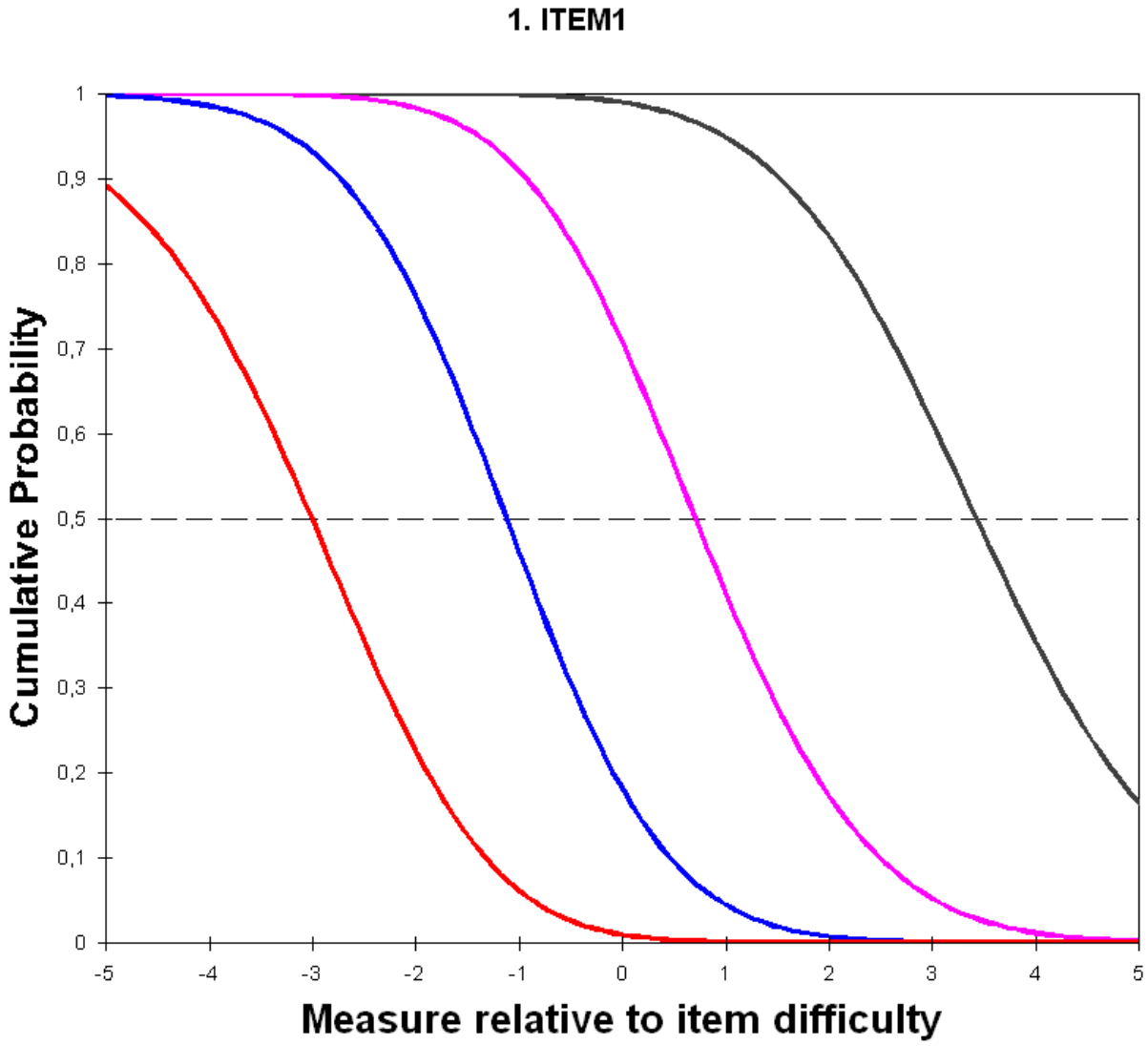


Abb. 1.13 Cumulative Probabilities Curve



Kapitel 2
Scoright

Kathrin Gruber

Zusammenfassung Eine der Hauptannahmen der Item-Response-Theorie ist jene der lokalen stochastischen Unabhängigkeit, das heißt die Lösung eines Items ist unabhängig von der Lösung eines oder mehrerer Items desselben Tests. Doch diese kann unter gewissen Bedingungen verletzt sein. Dieses Problem versucht die Testlet-Response-Theorie durch das Bündeln abhängiger Items zu einem sogenannten „testlet" zu lösen. Dazu werden formale Item-Response-Modelle um einen Testletparameter erweitert. In komplexeren Modellen werden diese Abhängigkeiten zwischen den Item-, sowie den Testletparamteren zusätzlich durch die Schätzung einer Kovarianzmatrix modeliert. Mit dem Programmpaket SCORIGHT ist es möglich die Annahme der lokalen stochastischen Unabhängigkeit für solche Itembündel fallen zu lassen und auch andere Items auf die Verletzung dieser Annahme zu überprüfen.

2.1 Einleitung

Der Begriff "testlet"[1] stand ursprünglich bezeichend für eine Sammlung von Aufgaben die gemeinsam vorgegeben werden. Diese dienten dazu die Effizienz eines Tests um das Verständnis eines bestimmten Stimulus zu verbessern. Ein klassisches Beispiel für eine solche Testsituation sind u.a. Leseverständnisaufgaben. Einerseits ist es zu wenig informativ nur einzelne Frage zu einer 250 Wörter langen Textpassage zu stellen, andererseits ist es problematisch nach einer gewissen Zeit neue, schwierige Items zu formulieren. Dadurch ergibt sich das Problem, dass zu wenig unabhängige Information durch die generierten Fragen gewonnen werden kann, da die „within-testlet"-Abhängigkeiten diese schmälern. Aus diesem Grund wurden solche langen Stimuli mit einem Bündel von Fragen gepaart. Auf diese einfache Art und Weise konnten solche Tests in effizienter Weise gestaltet werden. In der *Testlet-Response-Theorie* (TRT) kam es jedoch, ab dem Zeitpunkt, zu einer Wende als das (heute sehr vielfältig angewandte) *computerisierte adaptive Testen* (CAT) umsetzbar wurde. Allerdings steht und fällt das Konzept des CAT mit dessen fundamentaler Annahme, dass die Itemkennwerte, unberücksichtigt ihres Kontextes, konstant bleiben (engl. „assumption of item fungibility", auch bekannt als *Kontexteffekte*). Kontexteffekte sind all jene Einflüsse, die ein Item dadurch erzeugt, das es in Beziehung zu einem anderen Item, des selben Tests, steht (z.B.: Itempositionseffekte oder Kreuzinformationen). Gerade im CAT ist der Kontext jedes Items, aufgrund der Tatsache, dass immer unterschiedliche Tests bearbeitet werden, für jede Person verschieden. Diese Effekte wurden in Zusammenhang mit dem CAT bisher nur wenig geprüft.

Einen alternativen Lösungsansatz dieser Probleme bietet die TRT. Ein Testlet besteht (wie bereits erwähnt) aus einer Gruppe von Aufgaben, die als Einheit entwickelt wurden und gemeinsam vorgegeben werden. Auch die Items innerhalb eines Testlets können verzweigt, adaptiv (durch hierarchische

WU Wien, Institut für Statistik und Mathematik • kathrin.gruber@wu.ac.at

[1] Da für diesen Begriff keine deutsche Übersetzung bzw. kein deutsches Pendant existiert, wird dieser im Folgenden, der Einfachheit halber, immer mit *Testlet* übersetzt.

Strukturierung) oder in aufsteigendem oder absteigendem Schwierigkeitsgrad vorgegeben werden. Das Ziel des Bündelns von Items ist es, die Wahrscheinlichkeit dieser nachteiligen Effekte zu reduzieren ohne dabei die Effizienz des adaptiven Tests zu verringern. Des weiteren passt die Teststruktur oft auch besser zu dem Konstrukt über welches man eine Aussage treffen möchte.

Zusammenfassend können Testlets also benutzt werden um die Bedenken bzgl. des Einzelfallcharakters von alleinstehenden Items zu reduzieren, um Kontekteffekte in adaptiven Designs zu reduzieren und um die Effizienz des Testens bei einem verlängerten Stimulus zu maximieren.

2.2 Klassische Testlet Response Modelle

Der Ansatz von Bock (1972) ist stark an *Rosenbaum's Theorem der Itembündel* orientiert. Die Grundidee dabei ist, dass nur jene Items die ein Testlet bilden überhöhte lokale Abhängigkeiten besitzen. Die Verrechnung dieser lokal stochastisch abhängigen Itembündel erfolgt über ein *Item-Response-Theorie Modell* (IRT-Modell) für mehrkategorielle Daten. Das Testlet wird als Einheit betrachtet und mehrkategoriell verrechnet. Auch *Bock's Modell* postuliert eine einzige zugrundeliegende latente Dimension θ und repräsentiert die Wahrscheinlichkeit einen bestimmten Score zu erreichen, als Funktion der latenten Verteilung θ. Die Iteminformationsfunktion für den Rohwert $x = 0, 1, \ldots, m_j$, für ein Testlet j wird in diesem Modell wie folgt definiert

$$\Pr(T_{jx} = m_j) = \frac{\exp(a_{jk}\theta + c_{jk})}{\sum_{k=0}^{mj} \exp(a_{jk}\theta + c_{jk})} \tag{2.1}$$

Unter den Items befinden sich $j = 1, \ldots, J$ Testlets. Jedes dieser Testlet beinhaltet m_j Items, so dass die Wahrscheinlichkeit im j-ten Testlet für eine mehrkategorielle Antwort $x_j = 0, 1, \ldots, m_j$ (2.1) beträgt. Das Modell ist in obiger Form jedoch nicht voll identifizierbar, weswegen man zwei zusätzliche Restriktionen: (1) $\sum_{k=0}^{m_j} a_{jk} = \sum_{k=0}^{m_j} c_{jk} = 0$ und (2) $\theta \sim N(0, 1)$ einführt. Stellt man die Parameter a und c, bei geordneten Kategorien, als zentrierte Polynome dar, so erhält man das *Graded Response Partial Credit Modell* (Samejima, 1969).

Sind die Items jedoch unabhängig, so können sie mit einem traditionellen IRT-Modell verrechnet werden. Dieses Vorgehen führt zu interpretierbaren Itemparametern und Standardfehlern sowie zu einem Maß für die Güte der Anpassung. Des weiteren existiert eine Testlet-Informationsfunktion welche explizit den Beitrag des Itembündels zum Gesamttest zeigt.

Wenn man nun jedoch mehr Information über das Testlet erhalten will, oder Ad-hoc ein Testlet innerhalb einer adaptiven computerisierten Testvorgabe konstruieren möchte so benötigt man einen alternativen Ansatz.

2.3 Bayes'sche Testlet Response Modelle

Einen solchen alternativen Ansatz bietet die *Bayes'sche Statistik*. So könnte zum Beispiel ein Itemauswahlalgorithmus einen Testlet-Stimulus aus einem Itempool, aufgrund dessen Inhalt, dessen psychometrischen Eigenschaften und der bisherigen Antworten der Testperson auswählen. Dieser Ansatz besitzt also sozusagen ein Gedächtnis.

Die Bayes'sche Interpretation des Wahrscheinlichkeitsbegriffes lässt zu, dass man den Wert unbekannter Konstanten schätzt. Vorwissen und Grundannahmen über diese werden in einer Wahrscheinlichkeitsverteilung zusammenfasst welche die *A-priori-Wahrscheinlickeit* bildet (engl. „prior"). Ein einfaches Beispiel für eine solche A-priori-Verteilung sind zum Beispiel Mittelwert und Varianz der angenommenen Verteilung über die Schwierigkeitsparameter. Danach wird versucht, anhand neuer Daten, diese Annahme an den wahren Wert (den Populationsparameter) anzunähern. Allerdings beinhaltet die Wahl des „priors" einen gewissen Unsicherheitsfaktor, denn diese muss nicht immer

korrekt sein. Dieses Konzept steht ganz im Gegensatz zum klassischen frequentistischen Wahrscheinlichkeitsbegriff, in welchem Wahrscheinlichkeiten mittels Häufigkeiten interpretiert werden.

2.3.1 Der Testletparameter

Um die Abhängigkeit durch das Testlet zu modellieren wird der lineare Prediktor t_{ij} durch den sogenannten Testletparameter $\gamma_{id}(j)$, welcher die „within-testlet" Kovariation beschreibt, erweitert

$$t_{ij} = a_j(\theta_i - b_j - \gamma_{id(j)}) \tag{2.2}$$

wobei hier a_j den Anstieg (die Diskrimination), b_j die Schwierigkeit des Items, $\text{logit}^{-1} = \log(\frac{x}{1-x})$, θ_i die latente Fähigkeit der Testperson und $\gamma_{id(j)}$ den Testleteffekt (die Interaktion) von Item j mit Person i welcher im Testlet d_j genestet ist, bezeichnen. Nun wird dieser Effekt in das formale 3PL-Modell als zusätzlicher Parameter eingebaut:

$$P(Y_{ij} = 1) = c_j + (1 - c_j)\text{logit}^{-1}(t_{ij}) \tag{2.3}$$

Wie gewohnt bezeichnet in (2.3), Y_{ij} die Antwort von Testperson i bzgl. eines Items j und c_j den Parameter der unteren Asymptote, welcher im 3PL-Modell auch als Rateparameter bekannt ist. Das 2PL-Modell wird hier als Spezialfall des 3PL-Modells aufgefasst, welcher resultiert wenn alle Parameter für die untere Asymptote $c_j = 0$ gesetzt werden. Laut Definition beträgt die „within-testlet" Kovariation $\gamma_{id(j)} = 0$ für alle unabhängigen Items.

Dies lässt sich natürlich auch auf den mehrkategoriellen Fall, mittels dem *Ordinal Response Modell*, verallgemeinern:

$$P(Y_{ij} = r) = \phi(d_r - t_{ij}) - \phi(d_{r-1} - t_{ij}) \tag{2.4}$$

d_r, d_{r-1} bezeichnen in (2.4) die Itemkategorienparameter bei mehrkategoriellen Items und ϕ die kummulative Verteilungsfunktion der Normalverteilung.

Um nun all diese Informationen über die Testperson, die Items und die Testlets zu kombinieren wird ein *hierarschiches Bayes'sches Modell* zur Schätzung der Modellparameter benutzt. Dabei handelt es sich um ein Bayes'sches statistisches Modell, dessen A-priori-Verteilung $\pi(\lambda)$ in bedingte Verteilungen $p_1(\lambda|\lambda_1), p_2(\lambda_1|\lambda_2), \ldots, p_n(\lambda_{n-1}|\lambda_n)$, sowie in die Randverteilung $p_{n+1}(\lambda_n)$ zerlegt werden kann. Dieses Modell erlaubt die Zerlegung der A-priori-Verteilung in einen Teil mit strukturellen und einen zweiten Teil mit subjektiven Inforamtionen um so weitestgehend uninformativ zu bleiben.

2.3.2 Die Parameterschätzung

Im Falle des 3PL-Modells lässt sich die Likelihoodfunktion $P(Y|\lambda)$ wie folgt anschreiben

$$P(Y_{ij} = 1|\lambda_{ij} = (\theta_i, a_j, b_j, c_j)) = \prod_{i=1}^{l}\prod_{j=1}^{N_i}(c_j + (1 - c_j)\text{logit}^{-1}(a_j(\theta_i - b_j))) \tag{2.5}$$

Die obige Likelihoodfunktion besitzt die unbekannten Parameter $\lambda = (\lambda_{ij})$ welche unter dem Bayes'schen Ansatz flexible A-priori-Verteilungen benötigen. Im Speziellen unterstellt man $P(\lambda|\Lambda)$ folgende Verteilung: $\theta_i \sim N(0, 1)$ und $(\log(a_j), b_j, \text{logit}(c_j)) \sim N_3(\mu = (\mu_a, \mu_b, \mu_c), \Sigma)$[3]. Mittelwert und Varianz der Verteilung von θ werden auf 0 und 1 fixiert um die Identifizierbarkeit des Models zu gewährleisten. Klassischerweise wird auch angenommen, dass die Parameter $\log(a_j)$ sowie $\text{logit}(c_j)$, für $a_j > 0$ und $0 \leq c_j \leq 1$, einer Normalverteilung folgen. Die Kovarianzmatrix Σ beinhaltet die Abhängigkeiten zwischen den Itemparametern. Um die Modellspezifikation zu vervollständigen muss

man noch eine *Hyperprior Verteilung*[2] $\pi(\Lambda)$ festlegen. Dafür benutzt man die konjugierten A-priori-Verteilungen für $\mu \sim N_3(0, \upsilon \times I_3)$ und $\Sigma \sim$ Inv.-Wishart(S, n_0) (dies ist die multivariate Entsprechung der χ^2-Verteilung), wobei υ eine sehr groß gewählte Zahl (z.B. 100) und I_3 die Identitätsmatrix (um die Kovarianzmatrix der A-priori-Verteilung zu diagonalisieren) darstellen. Umgekehrt wird n_0 möglichst klein gewählt, um so minimal informativ zu sein. Damit sind die Modellspezifikationen des Bayes'schen TRT-Modells abgeschlossen.

Wie bereits erwähnt, ist in der Bayes'schen Statistik das letzendliche Ziel, Rückschlüsse über eine Menge von A-posteriori-Verteilungen der Modellparameter (2.6) anhand von A-posteriori Zufallsziehungen zu treffen.

$$p(\lambda|Y) \propto \int p(Y|\lambda)p(\lambda|\Lambda)\pi(\Lambda)d\Lambda \tag{2.6}$$

Das Integral der A-posteriori-Randverteilung lässt sich jedoch nicht in geschlossener Form lösen. Deswegen bedient man sich bei der Berechnung der Hilfe eines *Markov-Chain-Monte-Carlo* (MCMC) Verfahrens. Dazu definiert man eine *Markov-Kette*[3] - damit jeder Zustand t_i des MCMC-Samplers nur von t_{i-1} abhängt - welche zu seiner stationären Verteilung konvergiert.

Im ersten Schritt wählt man (z.B. durch die Generierung von Zufallszahlen die der A-priori-Verteilung von θ_i folgen) einen Wert um die Markov-Kette zu starten. Klarerweise konvergiert die Markov-Kette schneller zu seiner stationären Verteilung, wenn man Startwerte wählt die in der Nähe des zu schätzenden, Paramters liegen. Im zweiten Schritt zieht man zufällig aus der bedingten Verteilung $p(\lambda^{(t)}|\Lambda^{(t-1)}, Y)$, wobei der Vektor $\lambda^{(t)} = (\theta_i, a_j, b_j, c_j)$ die Parameter des TRT-Modells beinhaltet. Hierfür benutzt man unter anderem den *Metropolis-Hastings-Algorithmus* (vgl. dazu Chip & Greenberg, 1995). Im dritten Schritt zieht man ein Sample aus der bedingten Verteilung $p(\Lambda^{(t)}|\lambda^{(t)}, Y)$, wobei die Hyperprior Verteilung $\pi(\Lambda)$ typischerweise konjugiert zu $p(\lambda|\Lambda)$ ist und einer gebräuchlichen Verteilung enstammt. Im letzten Schritt des MCMC-Algorithmus startet man den Zufallsgenerator für M Iterationen, wobei M' Iterationen das sogenannte „burn-in" darstellen. Das heißt, man geht von Schritt 1 bis zu jenem Punkt, an welchem nur noch aus der stationären Verteilung gezogen wird. Nachdem die Markov-Kette konvergierte, zieht man nochmals $M - M'$ zusätzliche Werte um statistische Rückschlüsse zu ziehen (z.B. die Berechnung von Mittelwerten, Varianzen, Quantilen und Konfidenzintervallen). Das „burn-in", also M', wird durch die Gelman & Rubin Methode (1992) bestimmt. Dabei wird die Konvergenz zur stationären Verteilung der Markov-Ketten mittels eines F-Test, welcher die „across-chain" gegen die „within-chain" Variation in den durchlaufenen Ketten testet, überprüft.

2.4 Das Programmpaket SCORIGHT

SCORIGHT (Version 3.0) ist ein kleines Programmpaket zur Verrechnung von Tests die aus ein- oder mehrkategoriellen Items bestehen. Ebenfalls implementiert sind Modelle für die Kombination dichotomer und mehrkategorieller Items, sowie für Modelle die Kovariaten beinhalten. Die Items können unabhängig, in Testlets oder in einer Kombination aus beiden vorliegen. Falls Testlets in den Daten vorhanden sind, so schätzt das Programm das Ausmaß der lokalen Abhängigkeiten und adjustiert die Schätzer dementsprechend. Die Schätzung der Modellparameter erfolgt unter dem beschriebenen Bayes'schen Ansatz und unter Verwendung der erläuterten MCMC Methoden welche die einfache Berechnung vieler wichtiger statistischer Kenntwerte erlauben und unter anderen Ansätzen nicht möglich wären.

[2] Falls die A-priori-Verteilung bis auf einen Parameter τ bestimmt ist, also $\pi(\theta) = \pi(\theta, \tau)$ gilt, dann nennt man τ einen Hyperprior.

[3] Eine Markov-Kette ist ein spezieller stochastischer Prozess dessen Ziel es ist, Wahrscheinlichkeiten für das Eintreten zukünftiger Ereignisse anzugeben. Durch die Kenntnis einer begrenzten Vorgeschichte sind ebensogute Prognosen über die zukünftige Entwicklung möglich sind wie durch die Kenntnis der gesamten Vorgeschichte des Prozesses.

2.4.1 Die Verwendung von SCORIGHT

Das Programm SCORIGHT läuft in einer DOS Umgebung und kann ganz einfach wie jedes andere Programm gestartet werden. Danach wird man aufgefordert eine Reihe von Fragen - durch die einfache Eingabe über die Tastatur - zu beantworten. Die Antworten dienen dazu die Eingabedatei, welche die Daten beinhaltet, und die Platzierung der Ausgabedateien zu spezifizieren. Nach dem Start erscheint das in Abb.2.1 abgebildete Eingabefenster.

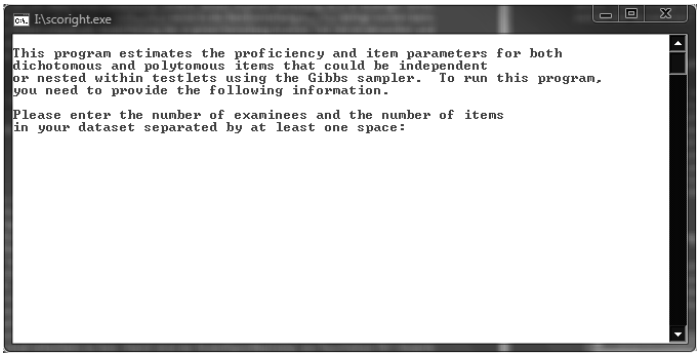

I:\scoright.exe
This program estimates the proficiency and item parameters for both
dichotomous and polytomous items that could be independent
or nested within testlets using the Gibbs sampler. To run this program,
you need to provide the following information.
Please enter the number of examinees and the number of items
in your dataset separated by at least one space:

Abb. 2.1 Das Eingabefenster nach dem Start von SCORIGHT inkl. Kurzinformation und der ersten Eingabeaufforderung.

Die Benutzung des Programms soll nun anhand eines simulierten Datensatzes, der aus 500 Personen und zwölf dichotomen Items besteht, illustriert werden. Des weiteren wurden zwei Testlets, zu jeweils vier und drei Items, eingebaut. Der Rest wurde als unabhängig spezifiziert. Im ersten Schritt wird man nun aufgefordert die Anzahl der Personen sowie der zu verrechnenden Items anzugeben. Bei der Dateneingabe ist zu beachten, dass Zahlen immer durch einen Abstand (einen oder mehrere Abstände oder Tabulatoren) separiert werden.

```
Please enter the number of examinees and the number of items in your dataset
separeted by at least one space: 500 12
```

Danach ist die Eingabe der Anzahl aller dichotomen Items, die im Datensatz enthalten sind, erforderlich. Falls keines der Items nur aus zwei Kategorien besteht, werden durch die Eingabe von 0 automatisch alle Aufgaben vom Programm als mehrkategoriell angesehen. In diesem Fall sind jedoch alle Items zweikategoriell.

```
Please enter the number of dichotomous items within the total 12 items: 12
```

Im nächsten Schritt wird verlangt, die Anzahl der 2PL Items (also ohne Rateparameter) im Datensatz zu spezifizieren. Im vorliegenden Beispiel beinhaltet keines der Items einen Rateparameter. Wäre die angegebene Zahl kleiner als die im vorherigen Schritt angebene Gesamtanzahl, so werden die restlichen Items von SCORIGHT automatisch als 3PL Items behandelt.

```
Please enter the number of 2PL binary response items: 12
```

Im vierten Schritt ist die Anzahl der in den Daten vorhandenen Testlets anzugeben. Falls jedoch alle Items unabhängig voneinander sind, ist diese Frage mit 0 zu beantworten. Der Beispieldatensatz enthält zwei Testlets.

```
Enter the total number of testlets in the test: 2
```

Mit der Folgenden Eingabeaufforderung definiert man den Dateipfad und die Datei welche die zu analysierenden Daten enthält.

```
Enter the name of the file that contains the test data: c:\subdirectory\Bsp1.dat
```

Es ist darauf zu achten, dass die Daten eine bestimmte Struktur aufweisen. Wie gewohnt bezeichnen Zeilen die Beobachtungen pro Person und Spalten die Items, welche von den Testpersonen bearbeitet wurden. Allerdings dürfen die Items durch keine Trennzeichen (Komma, Abstände usw.) separiert sein. Ebenfalls zu beachten ist, dass fehlende Werte in der Datenmatrix mit einem „N" kodiert werden. SCORIGHT behandelt diese fehlenden Werte damit als vollständig zufällig (engl. „missing completly at random").

```
000100101001
111010101101
011001010101
010001000000
010000100100
101111101111
010111111010
...
```

Die nächsten Schritte dienen dazu, nochmals die Position der zu verrechnenden Items und der Testlets zu spezifizieren. Das Datenfile enthält im vorliegenden Beispiel zwölf Items und 500 Personen, von denen alle in die Analyse miteinbezogen werden sollen. Die Items eins bis vier, sowie zehn bis zwölf, bilden die beiden Testlets.

```
Enter the starting and ending columns of the test scores for the data file: 1 12

Enter the starting and ending columns of Testlet #1: 1 4
Enter the starting and ending columns of Testlet #2: 10 12

Enter the starting and ending rows of the test scores: 1 500
```

In Folge ist es auch notwendig, eine Datei zu definieren und einzulesen, welche Informationen darüber enthält, ob die Items einen Rateparameter enthalten oder nicht (also 3PL oder 2PL) bzw. mehrkategoriell zu verrechnen sind. Das Zeichen „D" steht dabei für dichotome 3PL, „2" für dichotome 2PL und „P" für mehrkategorielle Items. Die Datei muss so strukturiert sein, dass die Definition für den Itemtyp steht in der ersten Spalte und die zugehörige Anzahl der Antwortkategorien in der zweiten Spalte steht. Die Iteminformationsdatei wird ebenso wie die Daten, auf die selbe Art und Weise eingelesen.

```
D 2
D 2
P 5
2 2
...
```

```
Enter the name of the item information file: c:\subdirectory\Iteminfo.dat
```

Da im Zuge der Schätzung und der Analyse sehr viele Outputfiles generiert werden, muss man ein Unterverzeichnis anlegen in welches die Ergebnisse der Analyse geschrieben werden.

```
Please enter the name of the subdirectory (include the last backslash) where you
want to put the analysis results, and make sure that there is no subdirectory
called "ch1", "ch2", ... under it: c:\subdirectory\result\
```

Nun muss man die Anzahl der Iterationen für den Sampling-Algorithmus festlegen. Da die Konvergenz von den Daten, sowie von den gewählten Startwerten beeinflusst wird, sind zumindest 4000 Iterationen empfehlenswert.

```
Enter the number of needed iterations of sampling: 4000
```

Es ist wichtig, dass der Sampling-Algorithmus konvergiert bevor gültige statistische Aussagen über das Modell getroffen werden können. In diesem Schritt spezifiziert der Benutzer also die Anzahl der vernachlässigbaren Ziehungen (das sog. „burn-in") aus den Iterationen für inferenzstatistische Zwecke. Für das vorliegende Beispiel wurden 3000 nicht zu berücksichtigende Ziehungen gewählt. Dies bedeutet, dass erst die Werte ab der 3001 Zufallsziehung in der Ausgabe aufgezeichnet und alle weiteren Berechnungen auf Grundlage dieser durchgeführt werden. Es ist jedoch ratsam, erst die Konvergenz der Markov-Ketten zu überprüfen und dann zu entscheiden welches „burn-in" adäquat ist.

```
Enter the number of draws to be discarded: 3000
```

Zusätzlich muss noch der Abstand zwischen den aufzuzeichnenden Ziehungen angegeben werden. Dies ist deswegen sinnvoll, da die Ziehungen aus der A-posteriori-Verteilung meist, aufgrund der Konstruktion der Markov-Kette, hoch korreliert sind. Durch die Wahl jeder k-ten Ziehung bleiben diese weitgehend unkorreliert und die Varianz der Schätzer kann ohne die Modellierung einer Zeitreihe berechnet werden. Für das simulierte Beispiel wird jede 11-te Ziehung in der Ausgabedatei aufgezeichnet.

```
Enter the size of the gab between posterior draws: 10
```

Im nächsten Schritt wird definiert wieviele Markov-Ketten man durchlaufen möchte. Dies soll dem Benutzer die Beurteilung der Konvergenz erleichtern. Das Programm SCORIGHT benutzt den F-Test von Gelman & Rubin (1992) als Konvergenzkriterium. Es ist jedoch zu beachten, dass je mehr Markov-Ketten man definiert, desto länger rechnet das Programm. Üblicherweise spezifiziert man zwischen drei und fünf Ketten.

```
How many chains do you want to run? 3
```

In Folge können noch Startwerte für die einzelnen Parameter - separat innerhalb jeder einzelnen Markov-Kette - angegeben werden. Auch diese müssen über eine Eingabedatei, welche wiederum eine bestimmte Struktur aufweisen muss, eingelesen werden. Dieses Vorgehen wird hier jedoch nicht weiter beschrieben. Der Anwender sei dafür jedoch auf das Benutzerhandbuch von Wang et al. (2005) verwiesen. In diesem Beispiel werden deswegen alle weiteren Fragen mit 0 beantwortet.

```
For CHAIN 1:
Do you want to input the initial values for item parameters a, b and c? If yes,
enter 1, otherwise enter 0: 0

For CHAIN 1:
Do you want to input the initial values for proficiency parameters theta? If yes,
enter 1, otherwise enter 0: 0

For CHAIN 2:
Do you want to input the initial values for item parameters a, b and c? If yes,
enter 1, otherwise enter 0: 0

For CHAIN 2:
Do you want to input the initial values for proficiency parameters theta? If yes,
enter 1, otherwise enter 0: 0

For CHAIN 3:
Do you want to input the initial values for item parameters a, b and c? If yes,
enter 1, otherwise enter 0: 0

For CHAIN 3:
Do you want to input the initial values for proficiency parameters theta? If yes,
```

```
enter 1, otherwise enter 0: 0


Do you have covariates for item parameters a (not including intercept)? If yes,
enter 1, otherwise enter 0: 0

Do you have covariates for item parameters b (not including intercept)? If yes,
enter 1, otherwise enter 0: 0

Do you have covariates for parameter theta? If yes, enter 1, otherwise enter 0:
0

Do you have any covariates for the testlet effects (not including intercept)?
If yes, enter 1, otherwise enter 0: 0
```

Nachdem alle benötigten Informationen angegeben wurden, gibt SCORIGHT am Ende nochmals eine kurze Übersicht über die eingegebenen Daten. So kann der Benutzer die Eingaben nochmals überprüfen bevor das Programm mit der Schätzung beginnt.

```
Please check the input:

-2 means independent items,
1 means the first testlet items,
2 means the second testlet items,
... and so on:

1111-2-2-2-2-2222

If the input is correct, enter 1, otherwise enter 0: 1
```

Sind alle Eingaben korrekt, beantwortet man die Frage mit 1 und SCORIGHT beginnt mit dem Durchlauf der Markov-Ketten. Am Bildschirm (siehe Abb. 2.2) erscheint zu Beginn die Startzeit jeder Kette, sowie die Zeit nach der jede 11-te Iteration beendet wurde. So erhält man Informationen darüber wieviel Zeit die Berechnungen benötigen werden. Im vorliegenden Beispiel benötigt SCORIGHT für 60 Iterationen nicht ganz 1 Sekunde. Für den Durchlauf einer Kette benötigt das Programm also in etwa 1 Minute. An dieser Stelle ist allerdings anzumerken, dass der simulierte Datensatz sehr einfach gestaltet ist. Kompliziertere Datensätze, die sowohl mehrkategorielle also auch 2PL und 3PL Items, sehr viele Testlets und Kovariaten in den Parametern enthalten, benötigen durchaus sehr viel längere Rechenzeiten.

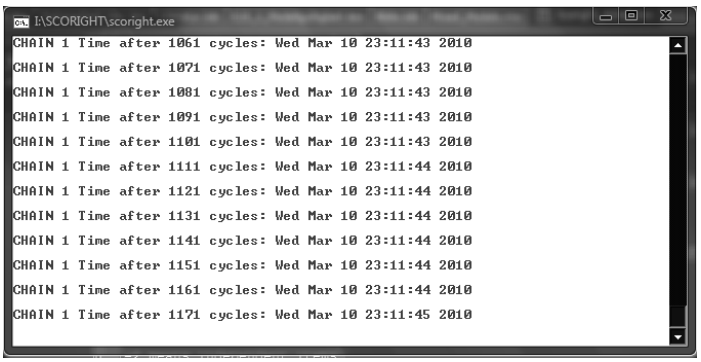

I:\SCORIGHT\scoright.exe
CHAIN 1 Time after 1061 cycles: Wed Mar 10 23:11:43 2010
CHAIN 1 Time after 1071 cycles: Wed Mar 10 23:11:43 2010
CHAIN 1 Time after 1081 cycles: Wed Mar 10 23:11:43 2010
CHAIN 1 Time after 1091 cycles: Wed Mar 10 23:11:43 2010
CHAIN 1 Time after 1101 cycles: Wed Mar 10 23:11:43 2010
CHAIN 1 Time after 1111 cycles: Wed Mar 10 23:11:44 2010
CHAIN 1 Time after 1121 cycles: Wed Mar 10 23:11:44 2010
CHAIN 1 Time after 1131 cycles: Wed Mar 10 23:11:44 2010
CHAIN 1 Time after 1141 cycles: Wed Mar 10 23:11:44 2010
CHAIN 1 Time after 1151 cycles: Wed Mar 10 23:11:44 2010
CHAIN 1 Time after 1161 cycles: Wed Mar 10 23:11:44 2010
CHAIN 1 Time after 1171 cycles: Wed Mar 10 23:11:45 2010

Abb. 2.2 Screenshot der Kurzinformationen des Iterationsprozesses des Sampling-Algorithmus beim Durchlauf der ersten Markov-Kette.

In dem im Vorfeld angelegten Unterverzeichnis werden von SCORIGHT die Ausgabedateien sowie einige zusätzliche Ordner ch1, ch2 und ch3 angelegt. Die Anzahl der Ordner entspricht der Anzahl der Markov-Ketten die durchlaufen wurden und enthalten die zufälligen Ziehungen des Sampling-Algorithmus. Da im Vorfeld weder Item-, Diskriminations- noch Personenparameter fixiert wurden, enthalten diese die Dateien „a_DrawsC", „b_DrawsC", „t_DrawsC", „SIGMA_DrawsC", „beta_DrawsC" und „gamV_DrawsC". Hierbei handelt es sich um die Ziehungen aus der A-posteriori-Verteilung für die Diskriminations-, Item- und Personenparameter, sowie der Kovarianzmatrix, der Koeffizienten der Kovariaten und der Varianz der Testlets. Die Größe dieser wird dabei durch die Anzahl der Iterationen (4000), die Länge des „burn-in" (1000) und den Abstand zwischen den Ziehungen (10) bestimmt. Im vorliegenden Beispiel enthalten die Dateien die Werte von 100 Ziehungen ((4000 − 3000)/10), sowie zwölf Spalten, was der Anzahl der analysierten Items entspricht.

In der Datei „itemP.est" sind sowohl die Schätzer der im Vorfeld spezifizierten Parameter der Items, als auch einige zusätzliche Informationen, enthalten. Der erste Teil der Datei enthält die Schätzer für die Item- und die Diskriminationsparameter sowie deren Standardschätzfehlern. In diesem Beispiel existieren jedoch keine Schätzer für die Rateparameter, da im Vorfeld nur 2PL Items definiert wurden. Wie man sieht, besitzen die Items hier alle einen Anstieg von in etwa eins, was auch konform mit den Daten ist, da diese nach dem dichotomen logistischen Modell von Rasch simuliert wurden.

Der zweite Teil der Datei enthält Informationen über die Koeffizienten der Kovariaten, sowie über die Schätzer für die Varianzen (SIGMA) und Kovarianzen (RHO) der Items. Wie bereits in Abschnitt 2.3.2 beschrieben beinhaltet die Kovarianzmatrix Σ die Abhängigkeiten zwischen den Itemparametern. Diese Implementierung ist einer der großen Vorteile von SCORIGHT, da nicht jede Software diese Abhängigkeiten berücksichtigt. Da es sich hier um dichotome Items handelt und im Vorfeld keine Kovariatenwerte für die Itemparameter spezifiziert wurden, erhält man jeweils nur den Koeffizienten für den Intercept und dessen Standardschätzfehler für die Kovariatenwerte der Item- und Diskriminationsparamter. Hätte man jedoch Kovariatenwerte für die jeweiligen Itemparameter angegeben, so würde man zusätzlich zum Intercept auch noch den Anstieg (bezeichnet mit „beta_1") erhalten. Diese beiden Kovariatenparamter sind jedoch nicht mit den eigentlichen Itemparamtern zu verwechseln, da sie eher im Sinne der linearen Regression zu interpretieren sind.

####	EST 'a'	SE('a')	EST 'b'	SE('b')	EST 'c'	SE('c')
1 2	1.0668	0.0939	-0.1545	0.1158	NA	NA
2 2	1.1779	0.1218	0.8390	0.1200	NA	NA
3 2	1.0540	0.1073	-1.0147	0.1308	NA	NA
4 2	1.1327	0.1124	-0.6800	0.1157	NA	NA
5 2	1.1572	0.1316	-0.5228	0.0976	NA	NA
6 2	1.0645	0.0900	-0.2574	0.1095	NA	NA
7 2	1.1005	0.0993	-0.5656	0.1067	NA	NA
8 2	1.0133	0.1061	-1.6091	0.1795	NA	NA
9 2	1.0254	0.0894	-0.2554	0.1103	NA	NA
10 2	1.0557	0.1489	-2.7241	0.3154	NA	NA
11 2	1.1346	0.1347	0.7323	0.1296	NA	NA
12 2	1.1405	0.1142	0.8259	0.1194	NA	NA

```
Estimated coefficients of 2-PL Binary item parameters:

For item parameter h (h=log(a)):
                        beta_0
Estimated values:       0.0829
s.e.:                   0.0610

For item parameter b:
                        beta_0
Estimated values:      -0.4472
s.e.:                   0.3333
```

```
Estimated covariance matrix of item parameters h(=log(a)) and b:
                      SIGMA_h    RHO_hb   SIGMA_b
Estimated values:      0.0130    0.0309    1.2443
s.e.:                  0.0139    0.0766    0.6598
```

Natürlich liefert auch SCORIGHT - wie jede andere IRT Standardsoftware - die geschätzten Personenparamter sowie deren Standardschätzfehler. Diese sind in der Datei „theta.est" enthalten. An dieser Stelle wird nur ein Auszug aus der Datei dargestellt.

```
####  EST Theta  SE(Theta)
   1    -0.4011     0.5440
   2     0.9565     0.6405
   3     0.7921     0.6486
   4    -0.5192     0.5788
   5     0.1343     0.5855
   6    -0.2616     0.5600
   7    -0.9897     0.5861
   8     0.6600     0.5834
   9     1.2637     0.7074
  10    -1.1857     0.5557
  11     0.4109     0.5982
  12    -0.9821     0.5882
  13    -0.0187     0.5972
  14    -0.3195     0.5569
  15    -0.5966     0.5536
  ...
```

Falls in den Daten Testlets enthalten sind und diese im Vorfeld definiert wurden, liefert SCORIGHT eine zusätzliche Datei („testlet.est") welche die geschätzte Varianz des Testletparameters γ für jedes Itembündel enthält. In diesem Beispiel sind die Varianzen beider Testlets klein was bedeutet, dass die Schätzer in der Nähe des wahren Parameters liegen und nur wenig streuen.

```
Estimated variance of the variance of the GAMMA for each testlet:

               Estimated      S.E.
Testlet   1:    0.6881      0.1834
Testlet   2:    0.6665      0.2440
```

Die Ausgabedatei „gamma.est" beinhaltet die geschätzten Testletparameter γ für jede Person. Da in diesem Beispiel zwei Testlets definiert wurden, erhält man zwei Spalten - also eine je Testlet - welche die Parameterschätzer für jede Person enthalten. Auch hier werden auszugsweise nur die Ergebnisse der ersten 15 Personen dargestellt.

```
   1  -1.5506   0.6555
   2   0.1397   0.0528
   3  -1.7678  -0.0669
   4  -0.2475   1.8680
   5   0.6290   0.9445
   6  -0.0382   0.8682
   7  -0.6466   0.2823
   8   1.1456  -1.8993
   9  -1.4698   0.2849
  10   0.2879   0.1598
  11   0.7793  -0.4612
  12  -0.6577   0.0460
```

```
13  -1.0941   1.0993
14  -1.2504   2.3548
15  -0.4695   0.4143
...
```

Da SCORIGHT die Methode nach Gelman & Rubin (1992) benutzt erhält man, nur wenn mehr als eine Markov-Kette durchlaufen wurde, die Datei ("convergence.est") welche Informationen zur Konvergenzdiagnose der kompletten Analyse enthält. Wurden in der Analyse Testlets berücksichtigt, so erhält man zusätzlich noch Informationen zur Konvergenz der Schätzungen der γ_i. Für jeden geschätzten Parameter gibt es zwei Statistiken zur Beurteilung der Konvergenz: „postshrink" und „confshrink". Die erste Statistik enthält die 2.5%, 50% und 97.5% Quantile der Zieldichte welche auf der Student-t-Verteilung basiert, und die zweite Statistik enthält die 50% und 97.5% Quantile einer groben oberen Schranke der Abschätzung darüber, inwieweit das Konfidenzintervall des A-posteriori Parameters sich verkleinern würde, wenn unendlich viele Iterationen durchlaufen würden. Laut Gelman & Rubin, sollten diese Werte kleiner als 1.2 sein um von Konvergenz der Ketten ausgehen zu können. Im hier behandelten Beispiel, konvergierten die Ketten für alle Werte bis auf den zweiten Testletparameter. Man sollte den Sampling-Algorithmus also nochmals mit mehr Iterationen durchlaufen um auch an dieser Stelle Konvergenz zu erzielen.

```
DIAGNOSIS FOR CONVERGENCE;

post: (2.5, 50, 97.5) quantiles for the target distribution
      based on the Student-t distribution

confshrink: 50and 97.5quantiles of a rough upper bound on
            how much the confidence interval of "post" will shrink
            if the iterative simulation is continued forever.

   If both components of confshrink are not near 1, the user
   should probably run the iterative simulation further.


2PL Binary Items:

Coefficients for Item Parameter a:
Beta_0
Post:
           -0.04            0.08            0.20
Confshrink:
            1.03            1.09

Coefficients for Item Parameter b:
Beta_0
Post:
           -1.10           -0.45            0.21
Confshrink:
            1.00            1.00

Variance Matrix of Item parameter a, and b:
Variance of a:
Post:
           -0.01            0.01            0.04
Confshrink:
            1.01            1.02
```

```
Covariance of a and b:
Post:
          -0.13           0.03           0.19
Confshrink:
           1.06           1.18

Variance of b:
Post:
          -0.07           1.24           2.56
Confshrink:
           1.02           1.05


Variance of Gamma for Testlet:

Testlet     1:
Posterior Range:
           0.32           0.69           1.06
Confidence Range:
           1.05           1.15

Testlet     2:
Posterior Range:
           0.14           0.67           1.19
Confidence Range:
           1.15           1.46
```

2.5 Diskussion

Das Programm SCORIGHT bietet mit seinem Bayes'schen Zugang eine echte Alternative zu den klassischen IRT-Softwarepaketen. Es ist von großem Vorteil, dass man so gut wie alle Ergebnisse der Schätzung erhält. Allerdings benötigt die Berechnung der Modellparameter in komplexeren Fällen, als den im obigen Abschnitt vorgestellten, sehr viel Zeit. Ein großer Nachteil des Programms ist auch, dass wenn einmal die Datenspezifikation vorgenommen wurde, man im Falle eines Fehlers diesen so einfach nicht mehr korrigieren kann. Der gesamte Fragenprozess muss erneut durchlaufen werden. Auch ist innerhalb des Programms keine grafische Aufbereitung der Ergebnisse möglich. Illustrationen der Itemparameter und der Verteilung der verschiedenen Personenparameter wären natürlich auch hier angebracht. Zwar könnte man diese Grafiken auch durch die Importierung der Ergebnisse in ein anderes Programm (z.B. in R) einfach selbst erzeugen, allerdings wäre eine automatische Generierung in SCORIGHT selbst um einiges praktischer. Dies könnte dann auch noch auf die Ziehungen des Sampling-Algorithmus erweitert werden indem die Ergebnisse jeder Iteration als Konfidenzbänder in einer Grafik dargestellt werden. Die Konvergenz der Markov-Ketten könnte auf diese Weise grafisch sehr einfach beurteilt werden.

Das Programm bietet jedoch eine gute Möglichkeit um Verletzungen der lokalen stochastischen Unabhängigkeiten in Testdesigns zu entdecken was in anderen Softwarepaketen nicht möglich ist.

Kapitel 3
FACETS Rasch Software illustriert am Beispiel von Multiple-Choice-Tests in Marketing

Margit Kastner, Amata Ring und Brigitte Stangl

Zusammenfassung Dieser Artikel soll in das Arbeiten mit der Software FACETS einführen und Beginnern bei den ersten Schritten der Verwendung dieses Rasch-Analyse-Programms begleiten. Dazu ist der Beitrag möglichst praxisnah geschrieben und theoretische Erläuterungen fallen eher kurz aus. Nach einem kurzen theoretischen Überblick über die Grundlagen des Multifacetten-Rasch-Modells liegt der Fokus auf der Durchführung eines realen Beispiels, um die Funktionsweise des Programms zu demonstrieren. Das Beispiel verwendet Daten einer Marketingprüfung der WU Wien und zeigt exemplarisch die Schätzung eines polytomen Rasch-Modells mit drei Facetten (Studenten, Fragen, Auswertungsschemen). Es wird gezeigt, wie die Schätzung mit dem Programm FACETS durchgeführt wird, was für Spezifikationen das Programm erlaubt und wie die Ergebnisse interpretiert werden können.

3.1 Einleitung

Bei der Erhebung von indirekt beobachtbaren Variablen wird die Evaluierung einer befragten Person als numerische Größe auf einer Beurteilungsskala ausgedrückt. Die numerischen Werte, welche auch als Rohwerte (raw scores) bezeichnet werden, fassen dann den Performance-/Fähigkeiten-Level eines Probanden zusammen und fließen in weiterer Folge in zu treffende Entscheidungen mit ein. Ein Problem, das sich dabei ergibt ist, dass die Rohwerte verzerrt sind. Verzerrungen ergeben sich etwa durch die Strenge des Beurteilers, die Schwierigkeit einer Frage/eines Items oder die Proband-Beurteiler-Item Interaktion. Ziel ist es, die nicht-linearen Rohwerte eines Probanden in ein lineares Maß umzuwandeln, korrigiert um Verzerrungen, damit einzig die Personenfähigkeit die Lösungswahrscheinlichkeit einer Frage beeinflusst (Linacre, 1993).

In diesem Zusammenhang ist die Arbeit des dänischen Mathematikers Georg Rasch (1960; 1980) richtungsweisend. Die Ansätze von Rasch haben großen Einfluss darauf, wie Forscher an die Messung von indirekt beobachtbaren Variablen herangehen. Messtheoretisch zählen Rasch-Modelle nicht zur klassischen Testtheorie sondern zur probabilistischen. Der Unterschied liegt darin, dass in der klassischen Testtheorie der Summenwert, der den Messwert eines Probanden darstellt, intervallskaliert ist. Bei probabilistischen Modellen sind auch bei Gültigkeit des Rasch-Modells Summenwerte nur ordinal skaliert. Demnach findet eine Überprüfung statt, ob einzelne Items als Maß der Fähigkeit einer Person zusammengefasst werden dürfen. Mittels probabilistischen Messmodellen können bei Kenntnis der Itemschwierigkeit und der Personenfähigkeit Verhaltensvorhersagen getroffen werden, d.h. es kann die Wahrscheinlichkeit prognostiziert werden, mit der ein Item gelöst wird (Bühner, 2006). Die Basis-Struktur des dichotomen Rasch-Modells ist in darauf aufbauenden Modellen wiederzufinden. Eine Weiterentwicklung ist das Multifacetten-Rasch-Modell (MFRM, Many-Facet Rasch Model), welches von Linacre im Jahr 1989 entwickelt wurde und das Linacre in der Windows-basierten Software

WU Wien, Institut für Tourismus und Freizeitwirtschaft • margit.kastner@wu.ac.at
WU Wien, Institut für Tourismus und Freizeitwirtschaft • amata.ring@wu.ac.at
WU Wien, Institut für Tourismus und Freizeitwirtschaft • brigitte.stangl@wu.ac.at

FACETS umgesetzt hat.

Bevor genauer auf das Rasch-Modell und das MFRM eingegangen wird, soll der Unterschied zwischen Rasch-Modellierung (Rasch modeling) und Rasch-Messung (Rasch measurement) kurz herausgestrichen werden. Laut Linacre (1994b) unterscheiden sich die beiden Ansätze grundsätzlich durch den Ansatz der Erweiterungen der ursprünglichen Arbeit von Rasch. Rasch-Messung entwickelte sich hauptsächlich um eine Forschungsgruppe von Ben Wright in Chicago. Rasch-Modellierung hingegen entwickelte sich in unterschiedlichen Forschungszentren wie z.B. in Österreich, Australien, Dänemark und anderen Ländern (von Davier, 2006). Mit dieser Unterscheidung im Hinterkopf wird in weiterer Folge genauer auf Rasch-Messung, insbesondere auf das MFRM eingegangen. Danach wird das Programm FACETS vorgestellt, wobei speziell für Anwender der Software relevante Aspekte diskutiert werden. Um theoretische Erläuterungen auf eine praktische Ebene zu bringen, wird schließlich ein Beispiel-Datensatz einer Marketing-Prüfung exemplarisch angewendet. Die Verwendung der Software bzw. die Modellspezifikation wird so Schritt für Schritt durchgespielt und die Herangehensweise an die Interpretation der Ergebnisse wird erläutet.

3.2 Rasch-Modell Grundlagen

Ziel einer Rasch-Analyse ist die Konstruktion von generalisierbaren, linearen Maßen, von denen man den Standardfehler (Reliability) und den Fit (Validität) kennt. Lineare Maße implizieren, dass ein Rating von zwei den gleichen Abstand zu drei hat wie ein Rating von drei zu vier. Das Vorhandensein dieser Struktur einer Ratingskala kann nur durch eine Überprüfung mittels empirischer Daten festgestellt werden (Linacre, 1994b). Rasch-Modelle betrachten Antwortmuster, wobei angenommen wird, dass die Antwortverteilung von den Eigenschaftsausprägungen abhängt (von Davier, 1997). Die Methode um verschiedene Parameter zu kombinieren ist additiv. Diese Form der Kombination ist möglich, weil bei Geltung des Rasch-Modells Eindimensionalität vorherrscht, d.h. die Items voneinander unabhängig sind. Demnach sind Antworten die Beobachtungen einer einzigen latenten Variable (Bejar, 1983). Dies impliziert, dass Parameter anhand einer gemeinsamen, linearen Skala ausgedrückt werden können (Linacre, 1994b). Folglich soll nur die Personenfähigkeit die Wahrscheinlichkeit der Lösung eines Items bzw. die Korrelation zwischen Items bestimmen und keine anderen Faktoren darauf Einfluss nehmen. Aufgrund dieser Eigenschaft werden Rasch-Modelle auch als Latent-Trait-Modelle bezeichnet und von Item-Response-Modellen abgegrenzt (Bühner, 2006).

Generell beschäftigen sich Rasch-Modelle damit, Maße für jeden Probanden zu bekommen, die frei von Verzerrungen sind. Das analytische Ziel ist die Transformation der einzelnen nicht-linearen Rohwerte in lineare Maße, korrigiert um Verzerrungen. Außerdem wird in probablistischen Modellen für jedes Maß eine lokale Schätzfehlervarianz berechnet, die am geringsten ist, wenn sehr viele Items verwendet werden und Item- und Personenparameter identisch sind (Linacre, 1993). Um das Ziel zu erreichen, wird eine probabilistische Beziehung zwischen der Fähigkeit einer Person und der Antwort auf ein Item angenommen. Dabei muss auch die Möglichkeit berücksichtigt werden, dass unfähigere Personen schwerere Items und fähigere leichtere Items lösen können – wenn auch die Wahrscheinlichkeit nicht sehr hoch ist. Um die Wahrscheinlichkeit zwischen null und eins zu berechnen, bedient man sich der logistischen Funktion, welcher im Kontext der IRT auch als Item-Characteristic-Curve (ICC) bezeichnet wird bzw. bei mehr als zwei Antwortkategorien als Category-Characteristics-Curve (CCC). So wird für jedes Item, losgelöst von der Schwierigkeit, je nach Fähigkeit einer Person, die Lösungswahrscheinlichkeit ermittelt. Die Werte der Items werden auf Basis der Randsummen der Daten (Person x Item) mittels joint- (uML), konditionaler- (cML) oder marginaler Maximum-Likelihood-Methode (mML) berechnet. Der Nachteil der uML ist, dass es durch die simultane Berechnung von Item- und Personenparametern zu Schätzproblemen kommen kann. Dieses Problem wird durch die Normalverteilungsannahme der Personenparameter bei mML reduziert. Am elegantesten ist allerdings cML, bei der nicht nur genanntes Schätzproblem vermieden wird sondern auch die Stichprobenzusammensetzung keinen Einfluss mehr auf die Itemparameterschätzung hat. Die Schätzung

erfolgt ohne dass Personenparameter im Voraus bekannt sein müssen, womit spezifische Objektivität (= unabhängige Schätzung der Item- und Personenparameter) garantiert wird, da Item- und Personenrohscores suffiziente Statistiken der jeweiligen Parameter sind. Die so errechneten Item- und Personenparameter besitzen dann die gleiche Einheit, wobei negative Werte auf einfachere Items oder unfähigere Personen hinweisen und vice verca. Mit diesen Kenntnissen kann man je Item direkt die Lösungswahrscheinlichkeit für eine Person prognostizieren (Bühner, 2006). Die Differenz zwischen Personen- und Itemparameter entspricht dem Logit der Lösungswahrscheinlichkeit (Fischer und Molenaar, 1995). Das Standard-Rasch-Modell schätzt dichotome Daten. Das erweiterte polytome Rasch-Modell verwendet Rating-Skalen oder Partial-Credit Beobachtungen und errechnet sich laut Andrich (1978b) und Masters (1982) wie folgt:

$$\log\left(\frac{P_{nik}}{P_{ni(k-1)}}\right) = B_n - D_i - F_k \tag{3.1}$$

P_{nik} Wahrscheinlichkeit, dass für Person n das Item i in Kategorie k beobachtet wird
$P_{ni(k-1)}$ Wahrscheinlichkeit, dass für Person n das Item i in Kategorie k-1 beobachtet wird
B_n Fähigkeit der Person n
D_i Itemschwierigkeit des Items i
F_k Barriere in Kategorie k beobachtet zu werden im Unterschied zu Kategorie k-1

So errechnete Summenwerte einer Person werden als erschöpfende Statistiken bezeichnet, weil sie alle Informationen enthalten um Aussagen über die Personenfähigkeit treffen zu können – Antwortmuster müssen somit nicht mehr beurteilt werden. Dies gilt aber nur bei Rasch-Konformität (Stelzl, 1993). Um nun zu ermitteln, welche Antwortkategorie ein Proband wählt, werden Schwellenwerte berechnet. Diese auch als Tresholds bezeichneten Werte liegen am Wendepunkt der logistischen Funktion. Liegt der Parameter einer Person rechts eines Tresholds ist die Wahrscheinlichkeit die höhere Antwortkategorie zu wählen größer. Bei k Kategorien werden immer $k-1$ logistische Funktionen sowie $k-1$ Tresholds geschätzt (Bühner, 2006).

Mittels Item-Response-Modellen kann auch die Trennschärfe von Items ermittelt werden. Je steiler der Anstieg einer ICC desto höher ist die Trennschärfe. Außerdem kann die Ratewahrscheinlichkeit einbezogen werden. [1] Beispielsweise hat eine Person bei einer Frage mit vier Antwortmöglichkeiten eine Ratewahrscheinlichkeit von 25 Prozent (Bühner, 2006).

Um Rasch-Modelle zu testen, gibt es verschiedene Ansätze: eine rein deskriptive Überprüfung ermöglicht der grafische Modelltest. Statistische Überprüfungen erlauben Likelihood-Quotienten-Tests, der Pearson-χ^2-Test oder die Cressie-Read-Statistik von Davier (1997). Da die Voraussetzungen für diese Tests selten erfüllt werden, bedient man sich häufig der Bootstrap-Methode, die eine Simulation der Prüfverteilungen ermöglicht. Modelltests überprüfen, ob die additive Verknüpfung von Personen- und Itemparameter gilt, ob die Summenwerte erschöpfende Statistiken sind, und ob spezifische Objektivität und Eindimensionalität vorliegen. Mittels des Likelihood-Quotienten-Tests kann durch Modellvergleiche jenes Modell gewählt werden, welches den Daten am besten entspricht. Der Pearson-χ^2-Test oder die Cressie-Read-Statistik prüfen auf signifikante Unterschiede zwischen beobachteten und erwarteten Antwortmustern (Bühner, 2006). Laut Rost (2004) erhält man mittels beider Tests ähnliche Ergebnisse. Um den Fit der Daten auf das Modell zu testen, werden die Beziehungen zwischen erwarteten und den beobachteten Fehler-Varianzen verwendet. Mittels Mean-Square-Fehlerstatistiken (Wright und Masters, 1982) – auch Infit- bzw. Outfit-Index genannt – werden die Abweichungen von den Erwartungen über alle Personen und Items zusammengefasst. Der Outfit ist eine χ^2-Statistik dividiert durch ihre Freiheitsgrade, dabei wird die Summe der standardisierten, quadrierten Fehler herangezogen. Der Infit vergleicht die Summe der quadrierten Rating-Fehler mit den erwarteten Werten (Rasch, 1960, 1980). Outfit-Werte reagieren sensibel auf Ausreißer und haben daher in Sachen Konsistenzfragen weniger Gewicht als Infit-Werte. Werte deutlich größer als eins zeigen, dass die

[1] Im Unterschied zu Item-Response-Modellen ist die Steigung bei Rasch-Modellen bei allen Items auf 1 fixiert. Erst Birnbaums Zwei Parameter Logistisches Modell (2PLM) bringt einen Diskriminationsparameter ins Spiel, der die Steigung modelliert. Das 3PLM fügt dem noch den Rateparameter hinzu.

Daten durch das Modell nicht gut prognostiziert werden können. Der Outfit und Infit kann Werte im Bereich zwischen 0 und $+\infty$ annehmen; der Erwartungswert liegt bei beiden bei 1. Akzeptable Werte befinden sich zwischen 0.5 und 1.5 (Linacre und Wright, 1994; Wright und Masters, 1990). In etwa gleiche Logits, aber mit unterschiedlichen Standardfehlern weißen auf die Präzision/Reliabilität der Maße hin. Ein schlechter Fit ist ein Hinweis dafür, dass keine Eindimensionalität vorherrscht (Linacre, 1994b). Die Reliabilität auch Test-Reliabilität genannt, misst die Genauigkeit mit der ein Test die Fähigkeit einer Person misst. Die Reliabilität ist jener Anteil der Stichprobenvarianz, der nicht auf Messfehler zurückzuführen ist. Reliablitätswerte bewegen sich zwischen eins und null (Eckes, 2003).

Die Erläuterungen zeigten, dass das finale Ziel einer Rasch-Analyse ein lineares Maß für den Grad der Fähigkeit eines Probanden ist. Zudem werden noch der Standardfehler und eine durch Fit-Statistiken quantifizierten Qualitätskontrolle ermittelt. Rasch-Maße generalisieren zu qualitativ ähnlichen, aber quantitative unterschiedlichen Situationen, z.B. durch hinzufügen von etwas schwereren Aufgaben bei einem Marketing Test, oder durch die Aufnahme von milderen Beurteilern (Linacre, 1993). Demnach sind geschätzte Parameter quantitative Darstellungen der zugrunde liegenden Variablen (z.B. Personenfähigkeit und Itemschwierigkeit) deren Bedeutung durch die qualitative Natur des Tests bestimmt wird. Parameter repräsentieren steigende Anteile der jeweiligen Variable, ausgedrückt etwa durch steigende Fähigkeit einer Person oder die steigende Itemschwierigkeit (Linacre, 1994b).

3.3 Multifacetten-Rasch-Modell

Wie auch das poltytome Rasch-Modell beschäftigt sich das MFRM damit, ordinale Rohwerte in lineare, objektive Maße zu verwandeln, von denen man die Standardfehler und Fit-Kriterien kennt. Für das MFRM gelten im Grunde die gleichen Prinzipien, die auch bereits im Abschnitt 3.2 aufgegriffen wurden. Das MFRM erweitert das polytome Rasch-Modell (siehe (1.1)) dahingehend, dass Verzerrungen durch die Strenge/Milde von Beurteilern/Judges berücksichtigt werden. Dadurch wird erreicht, dass Beurteilungen nicht nur genau sind, sondern auch so fair wie möglich (Linacre, 1994b).

$$\log\left(\frac{P_{nijk}}{P_{nij(k-1)}}\right) = B_n - D_i - C_j - F_k \tag{3.2}$$

P_{nijk} Wahrscheinlichkeit, dass für Person n das Item i bewertet durch den Beurteiler j in Kategorie k beobachtet wird

$P_{nij(k-1)}$ Wahrscheinlichkeit, dass für Person n das Item i bewertet durch den Beurteiler j in Kategorie k-1 beobachtet wird

B_n Fähigkeit der Person n

D_i Itemschwierigkeit des Items i

C_j Strenge des Beurteilers j

F_k Barriere in Kategorie k beobachtet zu werden im Unterschied zu Kategorie k-1

Demnach werden die im polytomen Rasch-Modell einbezogenen Facetten (Personen und Items) um eine weitere Facette ergänzt – den Beurteilern / Prüfern (Linacre, 1989, 1994b). Um diesen Faktor zu kontrollieren, wird für jede Person jene Leistung (performance) ermittelt, die bei einem durchschnittlich strengen Beurteiler erzielt worden wäre (Eckes, 2003). Es können allerdings beliebig andere bzw. weitere Facetten wie etwa Schwierigkeit der Aufgabe, verschiedene Auswertungsschemen, Muttersprache oder kultureller Hintergrund eingeführt werden, die die Leistung einer Person verzerren können. Besonders plakativ ist eine Verzerrung durch die Strenge von Beurteilern, weil Beurteiler als Experten handeln, die nach Ihrem Ermessen die Leistung bei Tests beurteilen. Deshalb wird in weiterer Folge die Idee des MFRM anhand der Facette 'Beurteiler' erklärt. Ziel ist, dass das Test-Ergebnis vom Wissen einer Person abhängen soll und nicht davon, ob er/sie einen strengeren oder milderen Beurteiler hat (Bond und Fox, 2007). Die genauesten Parameterschätzungen werden erzielt, wenn jeder Beurteiler alle Test bewerten würde. Allerdings ist das Rasch-Modell relativ robust gegenüber fehlenden Werten, d.h. bei Einbeziehung der Strenge von Beurteilern müssen nicht alle

Beurteiler alle Tests bewertet haben. Erforderlich ist lediglich, dass genügend Verbindungen zwischen allen Elementen [2] aller Facetten vorhanden sind, damit alle Parameter geschätzt werden können (Linacre, 1994b). Als Faustregel gilt, dass es ausreicht, wenn jeder Test von zwei Beurteilern begutachtet wurde (genaue Informationen dazu sind unter Linacre (1997) sowie Lunz et al. (1990) zu finden). Das MFRM von Linacre (1994b) ist so konzipiert, dass Facetten hinzugefügt werden können, aber es kann auch ein Standard Rasch-Modell gerechnet werden, d.h. mit zwei Facetten (Itemschwierigkeit und Personenfähigkeit).

Die Ergebnisse von MFRM werden im Facettenraum (siehe Abbildung 3.8) dargestellt, wodurch direkte Vergleiche zwischen den Maßen der einbezogenen Facetten möglich sind. Außerdem können Aussagen über die Konsistenz der Beurteiler gemacht werden. Erste Hinweise darauf können anhand der maximalen und minimalen Logitwerte ausgemacht werden. Eine Untersuchung der Fehlervarianzen ermöglicht es festzustellen, ob ein Beurteiler eher streng oder eher mild bewertet hat (Linacre, 1994b). Die Infit- und Outfit-Statistiken geben also die Schwankungen zwischen den Beurteilern an. Bezüglich der Beurteilerkonsistenz kann aus den Ergebnissen abgelesen werden, wie viele Infit- und Outfit-Werte der Beurteiler sich außerhalb des Intervalls von 0,5 bis 1,5 befinden. Weitere Kennzahlen sind die Reliabilitätskennzahlen, wobei hohe Reliabilität generell bedeutet, dass die gleiche Ordnung der Elemente der Facette auch mit anderen Daten abgebildet werden könnte (Bond und Fox, 2007) und der Separationsindex, welcher als das Verhältnis von 'wahrer' Streuung der Leistungsmaße (d.h. der Streuung der Leistungsmaße nach Standardfehlerkorrektur) zum 'durchschnittlichen' Standardfehler der Leistungsmaße ('Root Mean Square Error') ausgedrückt wird (Wright und Stone, 1999; Wright und Masters, 2002). Auf der Basis des Separationsindex kann der Index der Klassenseparation ('strata') berechnet werden (Wright und Masters, 2002). Dieser Index schätzt die Anzahl der potenziell unterscheidbaren Klassen einer Facette. Ein Index von 1 würde bedeuten, dass die Auswertungsschemen keine Unterschiede in ihren Strengemaßen aufweisen und damit austauschbar sind. Die Klassenseparation zeigt die Anzahl von statistisch reliabel unterscheidbaren Beurteilern auf. Dieser Wert wäre bei homogenen Beurteilern rund um eins. Die Separationsreliabilität hingegen, die auf die Genauigkeit der Unterscheidbarkeit der Strengewerte abzielt, geht bei homogener Strenge gegen null (Eckes, 2003).

3.4 Beschreibung von FACETS

In diesem Abschnitt soll Schritt für Schritt gezeigt werden, wie in FACETS ein MFRM geschätzt werden kann. Die Spezifikationen werden anhand des später detaillierter erläuterten Beispiels erklärt. Generell gesehen ist FACETS eine Erweiterung von WINSTEPS, wobei FACETS im Gegensatz zu WINSTEPS mit den Besonderheiten von MFRM umgehen kann. Das bedeutet, dass zusätzlich zu den beiden Facetten, die im Basis-Rasch-Modell enthalten sind (Personen und Items), weitere hinzugefügt werden können. Dies kann zum Beispiel die Facette 'Beurteiler' sein. Für diese Facette werden ebenso Parameter geschätzt. Es kann also verglichen werden, ob ein Beurteiler strenger oder milder als der andere bewertet. Die Schätzung von Parametern einer dritten (und weiterer) Facette ist in WINSTEPS nicht möglich. Einen vergleichenden Überblick über beide Programme zeigt Tabelle 3.1.

Öffnet man FACETS (siehe Abb. 3.1), erkennt man, dass es sich um ein Windows-basiertes Programm mit einigen der üblichen Windows-Menüfunktionen sowie diversen Erweiterungen handelt.

Grundsätzlich beginnt man mit der Modellspezifikation. Dazu besteht einerseits die Möglichkeit unter dem Menüpunkt 'Edit' eine Vorlage ('Edit from template') aufzurufen oder eine ganz neue Datei zu öffnen ('Edit new file'). Diese Modellspezifikation wird anschließend als Textdatei gespeichert und muss, um zur Ausführung zu gelangen, unter dem Menüpunkt 'Files' ('Specification File Name?') aufgerufen werden. Da es sich bei dem Spezifikationsdokument um ein normales Text-Dokument

[2] Eine einzelne Person, ein Item oder ein einzelner Beurteiler werden als Elemente bezeichnet. Im Unterschied dazu beziehen sich Facetten auf Personen, Items oder Beurteiler (Linacre, 1994b).

Tabelle 3.1 Vergleich WINSTEPS und FACETS

Merkmal	WINSTEPS	FACETS
Datentyp	rechteckig 2 Facetten: Personen und Items	2 Facetten: Personen, Items 3 Facetten: Personen, Items, Beurteiler 4 Facetten: Personen, Items, Beurteiler, Anlässe 5 Facetten: Paarvergleich und mehr ...
Datenkapazität	999.999 Personen bei 30.000 Items	999.999 Personen + Items + Beurteiler + ...
Auswertung	30 + Hauptabellen, Dateien, Diagramme, Graphen	8 + Hauptabellen, Dateien, Diagramme, Graphen
Konzeption und Interpretation	einfacher	komplexer
Rat	Verwende Winsteps wenn möglich z.B. Paarvergleiche, Leistungsbeurteilung	Verwende Facets wenn Winsteps die Arbeit nicht erledigt

Quelle: Winsteps (eigene Übersetzung)

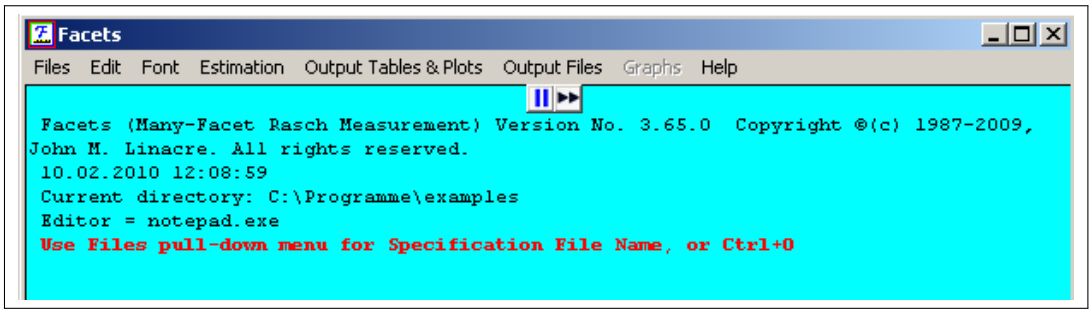

Facets
Files Edit Font Estimation Output Tables & Plots Output Files Graphs Help
Facets (Many-Facet Rasch Measurement) Version No. 3.65.0 Copyright ©(c) 1987-2009,
John M. Linacre. All rights reserved.
10.02.2010 12:08:59
Current directory: C:\Programme\examples
Editor = notepad.exe
Use Files pull-down menu for Specification File Name, or Ctrl+O

Abb. 3.1 Startfenster von FACETS

handelt, ist es auch möglich, das Dokument in jedem beliebigen Text-Editor zu erstellen und direkt über FACETS aufzurufen. Die Details der Modellspezifikation werden in Abschnitt 3.5.2 erklärt.

Nachdem FACETS immer eine Output-Datei (*.out) mit den Ergebnissen erzeugt, muss im nächsten Schritt der Name dieser Datei spezifiziert werden. Standardmäßig wird der gleiche Name wie bei der Datei mit den Modellspezifikationen vergeben. Danach fragt FACETS, ob man zusätzliche Spezifikationen angeben möchte (siehe Abb. 3.2). Für die meisten Analysen genügt es, dass Feld leer zu lassen und 'OK' zu klicken.

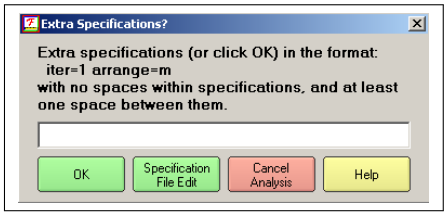


Abb. 3.2 Zusätzliche Spezifikationen

Da FACETS nun alle nötigen Informationen hat, beginnt es selbsttätig zu rechnen. Während der Parameterschätzungen informiert ein Fenster über den Fortschritt der Iterationen. Den Verlauf kann man auf dem Bildschirm mitverfolgen und es wird ein Auszug des Outputs, die ersten drei Tabellen, dargestellt. In Tabelle 1 findet man die wichtigsten, der vom Anwender festgelegten Modellspezifikationen (zur Erklärung der einzelnen Spezifikationen siehe Abschnitt 3.5.2). Tabelle 2 gibt einen Überblick über die Daten. In diesen beiden Tabellen kann also überprüft werden, ob FACETS die Daten und die Spezifikationen richtig übernimmt. In Tabelle 3 kann der Schätzvorgang mitverfolgt werden. Wenn die Parameterschätzung abgeschlossen ist, wird am unteren Ende des Bildschirms ange-

zeigt, welche Tabellen in der zuvor definierten Output-Datei erstellt wurden. Zusätzlich enthält diese Output-Datei auch die Tabellen 1 bis 3, die auch schon während der laufenden Parameterschätzung am Bildschirm zu sehen waren. Wie in Abbildung 3.3 zu sehen, wird in der letzten Zeile am Bildschirm der Pfad genannt in dem sich die Output-Datei befindet. Grundsätzlich öffnet sich diese Datei allerdings selbsttätig nach Abschluss der Analyse. Die vorletzte Zeile am Bildschirm enthält noch eine weitere wichtige Information, nämlich ob Untergruppen ('subsets') indentifiziert wurden oder nicht. Sollten keine identifierziert worden sein, so erscheint die Nachricht 'subset connection O.K.'; falls die Nachricht nicht erscheint, wurden Untergruppen in den Daten gefunden, für die nicht alle Informationen vorhanden sind. Das wäre bspw. dann der Fall, wenn ein Teil der Studenten nur von Beurteiler 1 und ein anderer Teil nur von Beurteiler 2 beurteilt worden wären, es aber keinen einzigen Studenten gibt, der von beiden Beurteilern (= Auswertungsschemen) bewertet wurde. Solche Untergruppen würden bedeuten, dass kein ausreichender 'Informations-Link' zwischen den Elemeten der Facetten besteht und somit ungenügende Information für die Schätzung der Beurteiler-Milde/Strenge vorhanden wäre (Bond und Fox, 2007). Linacre (1997) beschreibt die Mindestanforderungen, die erfüllt sein müssen, damit aus den Daten ein eindeutiges Messsystem geschätzt werden kann.

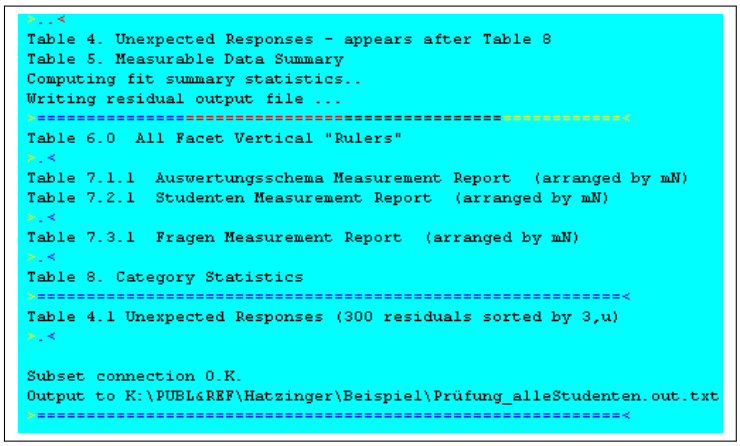


Table 4. Unexpected Responses - appears after Table 8
Table 5. Measurable Data Summary
Computing fit summary statistics..
Writing residual output file ...
Table 6.0 All Facet Vertical "Rulers"
Table 7.1.1 Auswertungsschema Measurement Report (arranged by mN)
Table 7.2.1 Studenten Measurement Report (arranged by mN)
Table 7.3.1 Fragen Measurement Report (arranged by mN)
Table 8. Category Statistics
Table 4.1 Unexpected Responses (300 residuals sorted by 3,u)
Subset connection O.K.
Output to K:\PUBL&REF\Hatzinger\Beispiel\Prüfung_alleStudenten.out.txt

Abb. 3.3 Informationen am Bildschirm

3.5 Exemplarische Darstellung einer Anwendung

Nachdem der technische Ablauf der Schätzung von FACETS dargelegt wurde, soll nun exemplarisch auf ein Beispiel eingegangen werden. Insbesondere wird die genaue Spezifikation des Modells, sowie die Interpretation des Outputs und der Graphiken, die vom Programm generiert werden, erläutert.

3.5.1 Daten

Grundlage der nachfolgenden Analysen bilden Daten einer Marketing-Prüfung an der WU Wien, welche im November 2008 stattfand. Der vorliegende Datensatz besteht aus 1118 Studierende, die zu dieser Prüfung angetreten sind.

Der Datensatz umfasst 31 Multiple-Choice-Fragen mit jeweils fünf Antwortmöglichkeiten, wobei jeweils eine oder mehrere richtig sein können. Der Fragen-Typ mit mehreren richtigen Antworten wird in der Literatur 'Multiple Correct Options' genannt und wurde 1953 von Dressel & Schmidt eingeführt. Der Vorteil dieser Fragen liegt darin, dass einerseits das Erraten der richtigen Lösung verringert wird. Andererseits besteht auch die Möglichkeit Teilwissen zu honorieren (Ben-Simon et al., 1997).

Zur Auswertung dieser 'Multiple Correct Options'-Fragen stehen mehrere Auswertungsschemen zur Verfügung. Die einfachste und gebräuchlichste ist die sogenannte 'Number of Rights' (NR) oder 'Number correct' (NC) Regel. Bei diesem Auswertungsschema bekommt man nur Punkte, wenn die Frage ganz richtig beantwortet wurde. Dieses 'Alles oder Nichts'-Auswertungsschema ist insbesondere für nicht so begabte Studierende von Nachteil (Reid, 1976), da es keine Punkte für teilweise richtig gelöste Fragen gibt (Ben-Simon et al., 1997; Bereby-Meyer et al., 2002). Daher schlägt Frary (1989) vor, alle richtig klassifizierten Antwortmöglichkeiten zu zählen. Dies bedeutet, dass es Teilpunkte für jede Frage gibt und somit auch 'Halbwissen' belohnt wird. Man kann diese zwei Auswertungsschemen quasi als zwei Beurteiler ansehen, bei denen einer nur ganz richtige Antworten als korrekt ansieht und somit fünf Punkte vergibt, während er 'halbrichtige' Antworten ignoriert und null Punkte vergibt, da er der Meinung ist, dass man beispielsweise in der Medizin mit 'Halbwissen' einen Menschen umbringen oder in der Wirtschaft eine Firma damit ruinieren könnte. Der zweite Beurteiler ist ein viel milderer Beurteiler und honoriert auch Teilwissen. In der Folge werden die beiden Auswertungsschemen 'Alles oder Nichts' und 'Teilpunkte' genannt. Tabelle 3.2 veranschaulicht die Markierungsinformationen (= Lösungen) einer Frage sowie die Punkte der Frage für beide Auswertungsschemen. Insgesamt gibt es bei dieser Frage drei richtige und zwei falsche Antwortalternativen. Person 1 hat beispielsweise alle drei richtigen Antwortmöglichkeiten und keine falsche Antwortmöglichkeit markiert und bekommt somit bei beiden Auswertungsschemen fünf Punkte. Person 2 hat nur zwei der drei richtigen Antworten markiert und bekommt beim Auswertungsschema 'Alles oder Nichts' null Punkte; bei der Bewertung 'Teilpunkte' bekommt die Person allerdings vier Punkte, da zwei richtige Antworten markiert wurden und die beiden falschen Antworten nicht markiert wurden, d.h. es wurde nur eine der fünf möglichen Antworten falsch beantwortet.

Tabelle 3.2 Vergleich der Auswertungsschemen: 'Alles oder Nichts' und 'Teilpunkte'

Lösung	Richtig	Richtig	Falsch	Richtig	Falsch	Punkte bei 'Alles oder Nichts'	Punkte bei 'Teilpunkte'
Person 1	x	x		x		5	5
Person 2	x			x		0	4
Person 3	x	x	x		x	0	3
Person 4	x		x	x	x	0	2
Person 5		x	x		x	0	1
Person 6			x		x	0	0

3.5.2 Modellspezifikation

Wie in Abbildung 3.4 ersichtlich, wird zu Beginn der Titel spezifiziert. Anschließend wird die Anzahl der Facetten genannt

```
Facets = 3
```

Der Strichpunkt (;) bedeutet, dass die Anmerkungen danach nicht für die Analysen verarbeitet werden. Hier können also sämtliche Kommentare und Erklärungen eingefügt werden. In Abbildung 3.4 sind so in Stichworten die jeweiligen Spezifikationen kurz erklärt.
Mit

```
Positive = 2
```

wird festgelegt, dass Facette 2 eine positive Orientierung hat. Das bedeutet, dass ein höherer Rohwert ein höheres Rasch-Maß (z.B. Personen- oder Itemparameter) erzeugt. Im konkreten Fall also: je mehr Fragen ein Student richtig hat, desto fähiger ist der jeweilige Student. Für die beiden anderen Facetten ist es genau umgekehrt. Für die Fragen heißt das beispielsweise: je höher der Rohwert desto niedriger

das Rasch-Maß. Dies ist gleichbedeutend mit: je mehr Studenten die Frage richtig beantwortet haben, desto leichter ist dieses Item. Durch

```
Noncentered = 1
```

wird festgelegt, dass für alle Facetten, außer für die erste, der Mittelwert auf 0 festgesetzt wird. Weiters kann

```
Inter-rater = 1
```

festgelegt werden. Dadurch werden für die Facette 1 in der Tabelle 7 der FACETS-Ergebnisdatei (siehe Abbildung 3.9) die Inter-Rater Reliabilitäten ausgegeben. Das macht vor allem dann Sinn, wenn tatsächlich (menschliche) Beurteiler die Elemente der Facetten sind. Es kann dadurch ein Hinweis bekommen werden, ob die Beurteiler wie Experten oder wie reine 'Beurteilungsmaschinen' handeln. Letzteres ist dann der Fall, wenn die beobachteten Prozent der exakten Übereinstimmung (Exact Agree. Obs%) größer als die erwarteten Prozent (Exp.%) sind. Es ist also nicht verwunderlich, dass wir im hier gerechneten Beispiel (siehe Abschnitt 3.5.3.1) ein solches Verhältnis finden, da die 'Beurteiler' ja Auswertungsschemen (und keine Menschen) sind, die per definitionem als 'Beurteilungsmaschinen' agieren.
Durch die deutlich höhere Anzahl an exakten Übereinstimmungen als erwartet, werden von allen Facetten die Standardfehler gegen 0 verzerrt. Durch $\sqrt{\frac{\text{observed exact agreements}}{\text{expected exact agreements}}}$, in diesem Fall $\sqrt{\frac{19609}{14739{,}7}}$, kann eine Maßzahl ausgerechnet werden, mit der alle Standardfehler des Modells multipliziert werden müssten, um dieser Verzerrung Rechnung zu tragen.

```
Usort = u
```

legt fest, wie die Tabelle 4 der FACETS-Ergebnisdatei (Table 4. unexpected responses) sortiert werden soll. Durch 'u' wird die Tabelle nach der absoluten Größe der standardisierten Residuen in absteigender Reihenfolge sortiert. Eine aufsteigende Sortierung würde durch den Großbuchstaben 'U' erreicht werden. Eine weitere Möglichkeiten stellt zum Beispiel die Sortierung nach Facetten dar. Es ist außerdem möglich, kombinierte Sortierregeln zu definieren sowie sich die Tabelle in unterschiedlichen Sortiervarianten ausgeben zu lassen. In dem hier verwendeten Beispiel wurde

```
Usort = 3,u
```

festgelegt, was bedeutet, dass die Tabelle vorrangig nach Facette 3 (Fragen) und nachranging nach der absoluten Größe der Residuen in absteigender Reihenfolge sortiert wird. Aus dieser Sortierung ist ersichtlich, bei welchen Fragen die meisten unerwarteten Antworten auftreten. Sollten unterschiedliche Sortiervarianten gewünscht werden, ist dies möglich, in dem man beispielsweise

```
Usort = (3,u),(1,u)
```

spezifiziert. Dies würde zusätzlich zur bereits beschriebenen Sortierung auch eine Sortierung nach Facette 1 (Auswertungsschema) und nachrangig nach der absoluten Größe der Residuen in absteigender Reihenfolge anzeigen.
Welche Werte als 'unexpected responses' ausgegeben werden sollen, wird über

```
Unexpected = 2
```

festgelegt. Wenn dieser Befehl in der Spezifikationsdatei enthalten ist, werden alle standardisierten Residuen genannt, deren Wert mindestens 2 beträgt. Die Standardeinstellung ist auf 3 festgelegt. In dem hier vorgestellten Beispiel wurde die Standardeinstellung beibehalten.
Über

```
T4maximum = 300
```

wird festgelegt, dass die 300 größten Residuen, die den zuvor unter 'Unexpected =' festgelegten Wert von 3 (= Standardeinstellung) überschreiten, in Tabelle 4 der FACETS-Ergebnisdatei ausgegeben werden. Wenn keine Spezifizierung erfolgt, werden (nur) die 100 größten Residuen ausgewiesen. In Abbildung 3.6 sieht man, wie die Befehle umgesetzt werden, die über Usort und T4maximum gegeben

```
Title = Prüfung – Vergleich der Auswertung 'Alles oder Nichts' und 'Teilpunkte'

Facets = 3                  ; 3 Facetten: Auswertungsschema, Studenten und Fragen
Positive = 2                ; für Facette 2 (Studenten) höherer Rohwert = höheres Rasch-Maß
Noncentered = 1             ; nur für Facette 1 (Auswertungsschema) wird der Mittelwerte nicht auf 0 fixiert
Inter-rater = 1             ; legt fest, für welche Facette die Interrater-Reliabilität ausgegeben werden soll
Usort = (3,u),(1,u)         ; sortiert die Residuen der Tabelle 4 'unexpected responses'
T4maximum = 300             ; legt fest, wie viele Residuen in Tabelle 4 ausgegeben werden sollen
Residual File = res.txt     ; erzeugt eine Residuen-Datei
Heading lines = yes         ; die Residuen-Datei hat eine Titelzeile
QM = no                     ; die Labels in der Datei haben keine Anführungszeichen
Vertical = 1A,2*,3A         ; zeigt für Tabelle 6 an, wie die Facetten dargestellt werden sollen
                            ; 1A/3A: von der ersten und dritten Facette (Auswertungschema und Fragen)
                            ; werden die Namen abgebildet
                            ; 2*: von der zweiten Facette (Studenten) wird die Verteilung abgebildet
T4maximum = 300             ; legt fest wie viele Residuen in Tabelle 4 ausgegeben werden sollen
Arrange = mN                ; zeigt wie Tabelle 7 angeordnet wird
Barchart = all              ; zeigt alle Möglichkeiten der 'graphical description of facet statistics' an
                            ; (Tabellen 6.1–6.3 für die jeweiligen Facetten)
Models=
?,?,?,R5                    ; Modell: 3 Facetten
*                           ; Ende der Modellspezifikation
Labels=
1,Auswertungsschema         ; Auswertungsschema ist Facette 1, analog zu den "Judges"werden
                            ; hier 2 Auswertungsschemen für MC-Fragen verwendet
1= Alles oder Nichts
2= Teilpunkte
*                           ; Ende der Beschriftung von Facette 1 (Auswertungsschema)
2, Studenten                ; Studenten sind Facette 2
1–1118
*                           ; Ende der Beschriftung von Facette 2 (Studenten)

3, Fragen                   ; Fragen sind Facette 3
1= Branchenattraktivität
2= Einkaufslabor
...
...
...
29= BCG-Portfolio
30= SWOT-Analyse
31= Marktdurchdringung
*                           ; Ende der Beschriftung von Facette 3


data=
1,1,1–31,0,100,100,100,100,100,100,0,100,100,100,0,100,100,100,100,100,100,100,100,100,
100,0,100,100,100,100,100,100,0,100,100,0,100,100,100,100
...
...
...
2,1118,1–31,100,0,0,0,100,100,100,100,100,0,100,100,100,100,100,100,100,0,100,100,100,
100,100,100,100,0,100,100,100,100,100,100,100,0,100,100,0
```

Abb. 3.4 Modellspezifikation

wurden.

Um einen Überblick über alle Residuen zu bekommen, ist es sinnvoll, sich eine Residuen-Datei erstellen zu lassen. Durch

```
Residual File = res.txt
Heading lines = Yes
QM = no
```

wird eine Text-Datei erzeugt, in der alle Residuen gelistet sind. Weiters wird spezifiziert, ob in der Residuen-Datei eine Titelzeile enthalten sein soll oder nicht (heading lines) und ob die Namen der Elemente mit oder ohne Anführungszeichen ausgegeben werden sollen (QM). Die Residuen-Datei kann auch als .csv erzeugt werden. Die Residuen-Tabelle muss allerdings nicht in der Spezifikationsdatei bereits angefordert werden; im Pulldown-Menü unter 'Output Files' kann die Tabelle ebenso nachträglich erzeugt werden.

```
Vertical = 1A,2*,3A
```

legt fest, wie die Rasch-Maße in Tabelle 6 der FACETS-Ergebnisdatei graphisch dargestellt werden sollen. Abbildung 3.8 zeigt, was diese Befehle für Auswirkungen auf die Darstellung haben. Pro Facette wird bestimmt, was in die Tabelle eingetragen werden soll. 'A' zeigt die Namen ('labels') der Elemente an und '*' zeigt die Verteilung dieser Facette an. Weitere Möglichkeiten der Darstellung sind beispielsweise die Anzahl der Elemente an einer bestimmten Position ('C') oder die Nummer des jeweiligen Elements ('N').

```
Arrange = mN
```

legt die Sortierung für die Tabellen 7 in der FACETS-Ergebnisdatei fest (siehe Abbildungen 3.9, 3.10, 3.11). Im hier definierten Fall wird vorranging nach dem Rasch-Maß in absteigender Reihenfolge, nachranging nach der Nummer des Elements sortiert. Auch hier wird die Richtung der Sortierung (absteigend – aufsteigend) durch Klein- bzw. Großbuchstaben fixiert. Weitere Sortiermöglichkeiten wären alphabetisch nach dem Namen ('A'), nach dem Fit ('F'), sowie nach der point-biserial Korrelation ('P').
Durch die Spezifizierung von

```
Barchart = All
```

werden alle in den FACETS-Ergebnistabellen 7 verfügbaren statistischen Beschreibungen der Rasch-Maße pro Facette graphisch dargestellt. Durch 'Yes' wird nur eine Auswahl graphisch dargestellt (Rasch-Maß (Logit), Infit und Outfit – sowohl unstandardisiert als auch standardisiert).

Das Herzstück des ganzen Modells wird über

```
Models = ?,?,?,R5
```

definiert. Im hier definierten Fall wird durch die drei Fragezeichen angezeigt, dass alle Elemente einer Facette mit allen Elementen der anderen Facetten interagieren dürfen. Durch 'R5' wird die Skala der Antworten festgelegt. In diesem Fall ist es eine Rating Skala von 0 bis 5. Weitere Skalen, die spezifiziert werden können, sind beispielsweise 'D' für dichotome Antworten, 'R' für eine Rating Skala von 0 bis 9 oder B100 für eine Rating Skala in Prozenten. Das Basis-Rasch-Modell würde also mit

```
Models = ?,?,D
```

spezifiziert werden. Beliebige andere Rating und Partial Credit Skalen können über den Befehl

```
Rating Scale =
```

erzeugt werden.
Bevor die Daten an FACETS übergeben werden können, müssen noch die einzelnen Facetten spezifiziert werden. Wie die gesamte Labels-Spezifikation durchgängig eingegeben wird, ist in Abbildung 3.4 ersichtlich. Hier wird die Spezifikation jeder einzelnen Facette einzeln erklärt.

```
Labels=
1, Auswertungsschema
1= Alles oder Nichts
2= Teilpunkte
*
```

Zuerst wird festgelegt, dass die erste Facette 'Auswertungsschema' heißt. Weiters bedeutet in den Daten eine '1', dass diese Daten dem Auswertungsschema 'Alles oder Nichts' zuzuordnen sind; '2' bedeutet 'Teilpunkte'. Mit '*' wird die Beschriftung der ersten Facette abgeschlossen.

```
2, Studenten
1–1118
*
```

Facette zwei sind Studenten, denen zur Anonymisierung der Daten die Nummern 1–1118 zugewiesen wurden.

```
3, Fragen
1= Branchenattraktivität
2= Einkaufslabor
...
31= Marktdurchdringung
*
```

Die dritte Facette sind die Fragen. Hier wird jeder Fragennummer auch ein Name zugeteilt. Für die vollständige Bezeichnung der Fragen siehe Abbildung 3.8.
FACETS hat nun alle notwendigen Informationen erhalten, wie das Modell geschätzt werden soll, es fehlen also nur noch die Daten. Daten können direkt in der Spezifikationsdatei eingetragen, oder aus einer externen Datei geladen werden. Unterstützt werden .txt, .xls(x), .rda(ta), .sdata, .sav, und .dta Dateien. Die Daten müssen mit den vorher spezifizierten Facetten und Labels übereinstimmen. In diesem Beispiel werden die Daten direkt in der Spezifikationsdatei an FACETS übergeben.

```
data=
1,1,1–31,0,5,...,5
1,2,1–31,5,0,...,0
...
1,1118,1–31,0,0,...,0
2,1,1–31,2,5,...,5
...
2,1118,1–31,3,1,...,4
```

Es wird also pro Facette spezifiziert, um welches Element es sich handelt. Die erste Zeile zeigt, dass im 1. Auswertungsschema (Facette 1) der erste Student (Facette 2) bei der ersten Frage null Punkte, bei der zweiten Frage fünf Punkte, ... und bei der 31. Frage auch fünf Punkte erreicht hat. Die letzte Zeile der Daten besagt, dass im 2. Auswertungsschema der 1118. Student bei der ersten Frage drei Punkte, bei der zweiten Frage einen Punkte und bei der 31. Frage vier Punkte erreicht hat. Mit 1–31 wird angezeigt, dass für insgesamt 31 Fragen Beurteilungen vorliegen.

3.5.3 Interpretation der Ergebnisse

Nachdem – wie in Abschnitt 3.4 gezeigt – das Programm gestartet wurde und die Parameterschätzung durchgeführt wurde, können nun die Ergebnisse, die FACETS liefert, betrachtet werden.

3.5.3.1 Output-Datei

In der Output-Datei wird zu Beginn in Tabelle 1 (Table 1. Specifications) die Spezifikation wiedergegeben. Inkludiert werden hier nicht nur die explizit in der Spezifikationsdatei festgelegten Bestimmungen, sondern auch die Standardeinstellungen, die nicht über eigene Spezifikationen abgewandelt wurden (siehe Abbildung 3.5). Danach werden Tabelle 2 (Table 2. Data Summary Report) und Tabelle 3 der FACETS-Ergebnisdatei (Table 3. Iteration Report) reproduziert, die auch schon auf dem Bildschirm während der Parameterschätzung angezeigt wurden (siehe Abschnitt 3.4).

```
-------------------------------------------------------------------------------------------------
Table 1. Specifications from file "K:\PUBL&REF\Hatzinger\Beispiel\Prüfung_alleStudenten.txt".

Title = Prüfung -- Vergleich der Auswertung 'Alles oder Nichts' und 'Teilpunkte 25.02.2010 14:24:36
Data file = (K:\PUBL&REF\Hatzinger\Beispiel\Prüfung_alleStudenten.txt)
Output file = K:\PUBL&REF\Hatzinger\Beispiel\Prüfung_alleStudenten.out.txt
Residual output file = res.txt

; Data specification
Facets = 3
Non-centered =  1
Positive =  2
Labels =
 1,Auswertungsschema ; (elements = 2)
 2,Studenten ; (elements = 1118)
 3,Fragen ; (elements = 31)
Model = ?,?,?,R5,1

; Output description
Arrange tables in order = mN
Bias/Interaction direction = ability ; leniency, easiness: higher score = positive logit
Fair score = Mean
Pt-biserial = Measure
Heading lines in output data files = Y
Inter-rater coefficients reported for facet = 1
Omit unobserved elements = yes
Barchart = All
Total score for elements = Yes
T3onscreen show only one line on screen iteration report = Y
T4MAX maximum number of unexpected observations reported in Table 4 = 300
T8NBC show table 8 numbers-barcharts-curves = NBC
Unexpected observations reported if standardized residual >= 3
Usort unexpected observations sort order = 3,u
Vertical ruler definitions = 1A,2*,3A
WHexact -- Wilson-Hilferty standardization = Y

; Convergence control
Convergence = .5, .01
Iterations (maximum) = 0 ; unlimited
Xtreme scores adjusted by = .3, .5  ;(estimation, bias)
-------------------------------------------------------------------------------------------------
```

Abb. 3.5 Table 1: Modellspezifikationen

In Tabelle 4 (Table 4.1 Unexpected responses) sieht man die Studenten, die bei bestimmten Fragen unerwartet gut oder schlecht beurteilt wurden (siehe Abbildung 3.6). Das bedeutet, dass die Antwort so gar nicht den Erwartungen, sonstigen Leistungen bzw. Fähigkeiten entsprach. So hat beispielsweise Student 730 bei der Frage 'Branchenattraktivität' beim Auswertungssystem 'Alles oder Nichts' null Punkte erreicht, obwohl er gemäß seinen Fähigkeiten die Frage eigentlich hätte richtig beantworten müssen. Betrachtet man Student 228 und die Frage 'Einkaufslabor' so sieht man, dass der Student 5 Punkte beim Auswertungssystem 'Alles oder Nichts' erreicht hat. Seinen Fähigkeiten würde allerdings entsprechen, diese Frage nicht richtig zu. Nachdem es sich um Multiple-Choice-Fragen handelt, ist es natürlich auch möglich, dass der Student die Frage mit Glück richtig gelöst, also geraten hat.

```
-------------------------------------------------------------------------------------
 Table 4.1 Unexpected Responses (300 residuals sorted by 3,u).

 +-------------------------------------------------------------------------------------+
 | Cat  Score  Exp.  Resd StRes| N Auswertungsschema Num  Stud Nu Fragen               |
 |-----------------------------+-------------------------------------------------------|
 |  0     0     4.9  -4.9 -9.0 | 1 Alles oder Nichts  730 730   1 Branchenattraktivität |
 |  0     0     4.6  -4.6 -4.4 | 1 Alles oder Nichts  544 544   1 Branchenattraktivität |
 |  0     0     4.6  -4.6 -4.4 | 1 Alles oder Nichts  841 841   1 Branchenattraktivität |
 |  0     0     4.5  -4.5 -4.2 | 1 Alles oder Nichts  263 263   1 Branchenattraktivität |
 |  0     0     4.5  -4.5 -4.2 | 1 Alles oder Nichts  758 758   1 Branchenattraktivität |
 |  5     5      .4   4.6  4.5 | 1 Alles oder Nichts  228 228   2 Einkaufslabor         |
 |  0     0     4.6  -4.6 -4.4 | 1 Alles oder Nichts  640 640   2 Einkaufslabor         |
 |  0     0     4.6  -4.6 -4.4 | 1 Alles oder Nichts  719 719   2 Einkaufslabor         |
 |  0     0     4.6  -4.6 -4.4 | 1 Alles oder Nichts  867 867   2 Einkaufslabor         |
 |  5     5      .4   4.6  4.3 | 1 Alles oder Nichts  368 368   2 Einkaufslabor         |
 |  5     5      .2   4.8  5.8 | 1 Alles oder Nichts  162 162   3 Segmentierung         |
 |  5     5      .3   4.7  5.4 | 1 Alles oder Nichts  543 543   3 Segmentierung         |
 |  5     5      .3   4.7  5.3 | 1 Alles oder Nichts 1047 1047  3 Segmentierung         |
 |  5     5      .3   4.7  5.2 | 1 Alles oder Nichts  610 610   3 Segmentierung         |
 |  1     1     4.7  -3.7 -4.8 | 2 Teilpunkte         186 186   3 Segmentierung         |
 |  5     5      .3   4.7  4.7 | 1 Alles oder Nichts  294 294   3 Segmentierung         |
 |  5     5      .3   4.7  4.6 | 1 Alles oder Nichts 1014 1014  3 Segmentierung         |
 |  5     5      .4   4.6  4.3 | 1 Alles oder Nichts  784 784   3 Segmentierung         |
 |  5     5      .4   4.6  4.3 | 1 Alles oder Nichts  802 802   3 Segmentierung         |
```

Abb. 3.6 Table 4: Unexpected responses

Tabelle 5 (Table 5. Measurable Data Summary) zeigt einen kurzen Überblick über die Daten (siehe Abbildung 3.7). Es werden der Mittelwert über alle Beobachtungen (68.944), die Standardabweichung, wenn die Stichprobe als gesamte Population betrachtet wird (S.D. (Population)), sowie die Standardab-

weichung, wenn die Stichprobe als Stichprobe der Population angesehen wird (S.D. (Sample)) für den beobachteten und den erwarteten Wert, sowie der Residualwert und der standardisierte Residualwert angegeben.[3] Weiters können Informationen bezogen auf den Fit des Modells abgelesen werden (siehe Abschnitt 3.5.3.2).

```
-------------------------------------------------------------------------------
Table 5. Measurable Data Summary.

+-----------------------------------------------+
| Cat  Score  Exp.  Resd StRes|                 |
|-----------------------------+-----------------|
| 3.47  3.47  3.47   .00  .00 | Mean (Count: 68944)|
| 2.06  2.06  1.32  1.58  .98 | S.D. (Population)  |
| 2.06  2.06  1.32  1.58  .98 | S.D. (Sample)      |
+-----------------------------------------------+
Data log-likelihood chi-square = 141424.3125
Approximate degrees of freedom = 1147
Chi-square significance prob.  = .0000
                                      Count   Mean   S.D.   Params
Responses used for estimation      =     68944   3,47   2,06     1147
Responses in one extreme score     =       372   5,00   0,00        6
All Responses                      =     69316   3,48   2,05     1153
Count of measurable responses        = 68944,00
Raw-score variance of observations   =   4,23 100.00%
Variance explained by Rasch measures =   1,72  40,65%
Variance of residuals                =   2,51  59,35%
-------------------------------------------------------------------------------
```

Abb. 3.7 Table 5: Zusammenfassung der Daten

Einen guten Überblick über die Messergebnisse der Facetten erhält man in Tabelle 6 der FACETS-Ergebnisdatei (Table 6.0 All Facet Vertical 'Rulers'). Dieser in Abbildung 3.8 dargestellte Facettenraum ist in vier Spalten unterteilt. Die erste Spalte enthält die Rasch-Maße (Logits) und gibt das Fähigkeitsmaß der Studierenden, die Strenge/Milde des Auswertungsschemas und den Schwierigkeitsgrad der Fragen wieder. In der zweiten Spalte ist zu erkennen, dass das Auswertungsschema 'Teilpunkte' leichter ist (bzw. milder beurteilt) als das Auswertungsschema 'Alles oder Nichts'. Diese Interpretation ergibt sich aus der Festlegung, dass die Facette 'Auswertungsschema' negativ definiert ist (siehe Abschnitt 3.5.2 und Abbildung 3.4), wodurch festgelegt wird, dass für diese Facette das Rasch-Maß niedriger ist, je höher der Rohwert ist. Die dritte Spalte zeigt die Verteilung der Parameterschätzungen in Hinblick auf die Leistungsfähigkeit der Studierenden. Fähigere Studierende finden sich im Facettenraum weiter oben, unfähigere weiter unten. In der Abbildung stellt jedes Sternchen 13 Studierende dar. In der vierten Spalte sind die Fragen dargestellt, wobei schwierigere Fragen weiter oben und leichtere Fragen weiter unten im Facettenraum dargestellt sind.

In Tabelle 7.1.1 der FACETS-Ergebnisdatei (Table 7.1.1 Measurement Report), welche in Abbildung 3.9 ersichtlich ist, ist dieses Ergebnis auch anhand der Zahlen nachvollziehbar. Der 'observed average score' für 'Teilpunkte' ist deutlich höher (4,1) als für 'Alles oder Nichts' (2,8). Das resultiert in einem Rasch-Maß von -0,09 für 'Alles oder Nichts' und von -0,60 für 'Teilpunkte'. Diese Parameter sind auch die Positionen, die in Tabelle 6 (Abbildung 3.8) abgebildet werden. Äquivalent dazu kann auch die leichteste Frage identifiziert werden, nämlich 'Barter Geschäfte'.

Da die Lage der beiden Facetten 'Studenten' und 'Fragen' laut der verwendeten Modellspezifikation fixiert ist (siehe Abschnitt 3.5.2 und Abbildung 3.4), kann die Lage der Facette 'Auswertungsschema' relativ zu den beiden anderen Facetten gesehen werden. Generell trifft das Auswertungsschema 'Alles oder nichts' in etwa die Fähigkeit der Studenten (Mittelwert nahe null), 'Teilpunkte' ist eher zu leicht (deutlich niedrigerer Parameter).
Die weiteren Tabellen, die im Output enthalten sind, beziehen sich größtenteils auf den Fit des Modells und werden nun gesondert betrachtet.

[3] Die Werte unter Cat bzw. Score sind nur dann unterschiedlich, wenn bspw. dichotome Daten mit 1 und 2 kodiert wären und sie im Zuge der Schätzung des Modells auf 0 und 1 umkodiert werden müssten. Dann wäre 'Score' der Wert, der zu den 0 und 1 kodierten Daten gehört.

```
Table 6.0  All Facet Vertical "Rulers".

Vertical = (1A,2*,3A,S) Yardstick (columns lines low high extreme)= 0,10,-1,2,End
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------+
|Measr|-Auswertungsschema|+Studenten |-Fragen                                                                                                                              |
|-----+------------------+-----------+-------------------------------------------------------------------------------------------------------------------------------------|
|   2 +                  + .         +                                                                                                                                     |
|     |                  |           |                                                                                                                                     |
|     |                  |           |                                                                                                                                     |
|     |                  |           |                                                                                                                                     |
|     |                  |           |                                                                                                                                     |
|     |                  |           |                                                                                                                                     |
|     |                  |           |                                                                                                                                     |
|     |                  |           |                                                                                                                                     |
|     |                  | .         |                                                                                                                                     |
|     |                  | .         |                                                                                                                                     |
|   1 +                  + .         +                                                                                                                                     |
|     |                  | .         |                                                                                                                                     |
|     |                  | **        |                                                                                                                                     |
|     |                  | *.        |                                                                                                                                     |
|     |                  | *.        |                                                                                                                                     |
|     |                  | ***.      |                                                                                                                                     |
|     |                  | *****.    | Positionierung       Segmentierung                                                                                                  |
|     |                  | ******    | Break-Even-Analyse   Distributionsstufen   Einkaufslabor                                                                            |
|     |                  | ********. | BCG-Portfolio        Marktdurchdringung    Maslow               Partial-/Totalm.                                                  |
|     |                  | **********| Adoptionsprozess     Branchenattraktivität Kennzahlen           Verbundsysteme    Zeitl. Abstimmung                       |
*   0 *                * ********. * Datenerhebung       Differenzierung       Kennzahlen/Interpr.  Marktziele        Werbestil          Wirtsch.-Aanlayse *
|     | Alles oder Nichts| ********. | Marketing-Aufgaben   SWOT-Analyse          Verkaufsförderung                                                                    |
|     |                  | ********. | Delphimethode        Konfidenzintervall    Skimming Strategie                                                                   |
|     |                  | *****.    | Geschl. Frage        Involvement                                                                                                |
|     |                  | ******.   | Markoumfeld-Analyse  Mystery Shopping                                                                                           |
|     |                  | *****.    |                                                                                                                                     |
|     | Teilpunkte       | ***.      |                                                                                                                                     |
|     |                  | *.        | Barter-Geschäft                                                                                                                     |
|     |                  |           |                                                                                                                                     |
|     |                  |           |                                                                                                                                     |
|  -1 +                  +           +                                                                                                                                     |
|-----+------------------+-----------+-------------------------------------------------------------------------------------------------------------------------------------|
|Measr|-Auswertungsschema| * = 13    |-Fragen                                                                                                                              |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------+
```

Abb. 3.8 Table 6: Übersicht über die Facetten

3.5.3.2 Fit des Modells

Generell ist in diesem Beispiel der Fit des spezifizierten Modells mit den Daten nicht vorhanden. Dies wurde allerdings auch nicht erwartet, da klar ist, dass das Auswertungssystem 'Alles oder Nichts' strenger beurteilt als das Auswertungssystem 'Teilpunkte'. Ein erster Überblick kann gewonnen werden, wenn man sich das Residuen-File genauer ansieht. Erwartet wird, dass ca. 1 % der standardisierten Beobachtungen außerhalb von +/- 3 und 5 % außerhalb von +/- 2 liegen. Im hier beschriebenen Beispiel liegen die Werte mit 1,4 % der standardisierten Residuen außerhalb von +/- 3 und 4,6 % außerhalb von +/- 2 knapp über bzw. unter den Referenzwerten.
Einen weiteren Anhaltspunkt für den schlechten Fit liefert die Tabelle 5 der FACETS-Ergebnisdatei (siehe Abbildung 3.7). Hier werden der χ^2-Wert und die damit verbundene Signifikanz angegeben, die anzeigen, dass das Modell nicht fittet. Weiters kann der Anteil der Varianz, der durch das Rasch Modell erklärt wird, abgelesen werden. Auch diese ist in diesem Fall mit knapp über 40 % relativ niedrig.

Die Tabellen 7.1.1 bis 7.3.1 der FACETS-Output-Datei (siehe Abbildungen 3.9, 3.10, 3.11) liefern die Ergebnisse für die einzelnen Facetten. In allen Tabellen werden zuerst die Werte für jedes Element der Facette ausgegeben. Für die Facette 'Auswertungsschema' sind dies also 'Alles oder Nichts' und 'Teilpunkte', für die Facette 'Studenten' werden die Werte für alle 1118 Studenten angezeigt und für die Facette 'Fragen' für alle 31 Fragen. Um den Fit zu beurteilen, können Infit und Outfit Mean Square betrachtet werden. Der Outfit ('outlier-sensitive fit') gibt an, inwieweit ein normalerweise konsistent einstufender Beurteiler unerwartete Bewertungen am äußeren Ende der Skala abgibt. Der Infit 'inlier-sensitive fit' oder 'information-weighted fit') reagiert hingegen empfindlich beim unerwarteter Beurteilungen im mittleren Skalenbereich. Wie man den Abbildungen 3.9 und 3.11) entnehmen kann, liegen sowohl Infit als auch Outfit innerhalb der Referenzwerte von 0,5 und 1,5. Bei Abbildung 3.10 erkennt man, dass die Werte im Mittel zwar im Referenzbereich liegen, die einzelnen Studenten allerdings davon abweichen. Dies kann sicherlich auch auf das Rateverhalten von Studenten bei Prüfungen zurückgeführt werden, da eine einzelne Frage nie zu negativen Punkten führt und es daher besser ist, irgendeine Antwortalternative anzukreuzen, wenn man die Frage nicht weiß und so vielleicht durch Glück doch Punkte zu erzielen. Am unteren Ende der Tabelle sind Mittelwerte und Standardabweichungen (analog zu Table 5, siehe Abbildung 3.7) über die gesamte

Facette angegeben. Im Hinblick auf das Auswertungsschema gibt der Separationsindex an, wie verlässlich zwischen diesen Schemen anhand ihrer Strengemaße unterschieden werden kann. Analog liefert der Separationsindex im Falle der Fragen Information über den Grad ihrer Unterscheidbarkeit anhand der Schwierigkeitsmaße. Wie bereits erwähnt kann auf Basis des Separationsindex der Index der Klassenseparation ('strata') berechnet werden. Im vorliegenden Fall ist kein Index von 1 gegeben, was bedeutet, dass die Auswertungsschemen Unterschiede in ihren Strengemaßen aufweisen und damit nicht austauschbar sind – dies wurde auch so erwartet. Bei den Studierenden kann man fünf verschiedene Klassen orten und bei den Fragen 23 unterschiedliche, was bedeutet, dass diese nicht austauschbar sind.

Weiters wird in diesen Tabellen die Reliabilität angegeben. Die Reliabilität von allen Facetten ist sehr hoch, was zum Beispiel für die Facette 'Auswertungsschema' bedeutet, dass mit einer anderen Stichprobe an Studenten und / oder Fragen auch dann wieder 'Alles oder Nichts' als die mildere Beurteilungsvariante identifiziert werden würde. Generell kann also geschlossen werden, dass diese Daten nicht mit einem MFRM mit drei Facetten abgebildet werden können.

Eine weitere Information, die aus diesen Tabellen gewonnen werden kann, ist die Überprüfung der Nullhypothesen, dass alle Auswertungsschemen gleich streng bzw. nachsichtig, alle Studenten gleich fähig, und alle Fragen gleich schwierig sind (Model, fixed (all same) chi-square). Diese Hypothese kann für alle drei Facetten verworfen werden, weil der χ^2-Wert für das Auswertungschema einen Wert in Höhe von 10557,3 bei einem Freiheitsgrad aufweist und die χ^2-Werte für die Studenten 15548 (1117 Freiheitsgrade) und für die Fragen 7674,1 (30 Freiheitsgrade) betragen.

```
-------------------------------------------------------------------------------------------------------------------------------
Table 7.1.1  Auswertungsschema Measurement Report  (arranged by mN).
+-------------------------------------------------------------------------------------------------------------------------------+
|  Total   Total   Obsvd  Fair-M|        Model | Infit      Outfit    |Estim.| Correlation | Exact Agree. |                    |
|  Score   Count  Average Avrage|Measure  S.E. | MnSq ZStd  MnSq ZStd|Discrm| PtMea PtExp | Obs %  Exp % | N Auswertungsschema |
|-------------------------------+--------------+---------------------+------+-------------+--------------+--------------------|
| 97745   34658       2.8   3.04|   -.09   .00 | 1.31  9.0  1.32  9.0| 1.42 |   .51   .54 |  56.9   42.8 | 1 Alles oder Nichts |
|143292   34658       4.1   4.48|   -.60   .00 |  .52 -9.0   .63 -9.0|  .61 |   .50   .41 |  56.9   42.8 | 2 Teilpunkte        |
|-------------------------------+--------------+---------------------+------+-------------+--------------+--------------------|
|120518.5 34658.0     3.5   3.76|   -.35   .00 |  .92   .0   .97   .0|      |   .50       |              | Mean (Count: 2)     |
| 22773.5      .0      .7    .72|    .26   .00 |  .39  9.0   .34  9.0|      |   .01       |              | S.D. (Population)   |
| 32206.6      .0      .9   1.02|    .36   .00 |  .55 12.7   .49 12.7|      |   .01       |              | S.D. (Sample)       |
+-------------------------------------------------------------------------------------------------------------------------------+
Model, Popuin: RMSE .00  Adj (True) S.D. .26  Separation 72.65  Strata 97.20  Reliability (not inter-rater) 1.00
Model, Sample: RMSE .00  Adj (True) S.D. .36  Separation 102.7  Strata 137.3  Reliability (not inter-rater) 1.00
Model, Fixed (all same) chi-square: 10557.3  d.f.: 1  significance (probability): .00
Inter-Rater agreement opportunities: 34472  Exact agreements: 19609 = 56.9%  Expected: 14739.7 = 42.8%
-------------------------------------------------------------------------------------------------------------------------------
```

Abb. 3.9 Table 7.1.1: Measurement Report für 'Auswertungsschema'

3.5.3.3 Graphiken

Die Graphiken werden einerseits unter Tabelle 8.1 der FACETS-Ergebnisdatei (Table 8.1 Category Statistics) dargestellt, andererseits besteht die Möglichkeit, die Graphiken unter dem Menüpunkt 'Graphs' aufzurufen. Grundsätzlich werden die 'category probability curves' (CPC) oder die Item-Charakteristik Kurve (ICC) angezeigt.

In Tabelle 8.1 der FACETS-Ergebnisdatei (siehe Abb. 3.12) werden zu Beginn die sechs Kategorien (0–5) dargestellt sowie die Häufigkeiten dieser. Hier zeigt sich eindeutig, dass hauptsächlich 0 oder 5 Punkte von den Studenten erreicht wurden, was auch auf das Auswertungssystem 'Alles oder Nichts' zurückzuführen ist. Weiters findet man in der Tabelle die Kategorienparameter ('measure category'). Diese können mit dem Personenparameter der Abbildung 3.10 verglichen werden. Man kann dabei erkennen, dass beispielsweise die Studenten 91 bis 1100 einen Parameter zwischen 3,72 und 0,93 haben. Nachdem dieser über dem Kategorienparameter von 0,92 liegt, wird erwartet, dass diese Studierende alle fünf Antwortalternativen als richtig erkennen. Betrachtet man hingegen die letzten Studierenden 990 bis 1060, so sieht man, dass der Personenparameter unter -0,71 ist, wodurch die Wahrscheinlichkeit, dass diese Studenten keine oder höchstens eine richtige Antwortalternative richtig erkennen, am höchsten ist. Weiters werden die Schwellenwerte oder Tresholds nach Rasch-Andrich

```
-------------------------------------------------------------------------------------------------------------
Table 7.2.1  Studenten Measurement Report  (arranged by mN).
+-------------------------------+-------------+-------------------+------+-------------+--------------------+
|  Total   Total   Obsvd  Fair-M|        Model | Infit      Outfit   |Estim.| Correlation |                    |
|  Score   Count  Average Avrage|Measure  S.E. | MnSq ZStd  MnSq ZStd|Discrm| PtMea PtExp | Num  Studenten     |
|-------------------------------+-------------+-------------------+------+-------------+--------------------|
|   310      62       5.0   5.00|(  3.72  1.79)|Maximum             |      |   .00   .00 |   91 91            |
|   310      62       5.0   5.00|(  3.72  1.79)|Maximum             |      |   .00   .00 |  420 420           |
|   310      62       5.0   5.00|(  3.72  1.79)|Maximum             |      |   .00   .00 |  679 679           |
|   310      62       5.0   5.00|(  3.72  1.79)|Maximum             |      |   .00   .00 |  702 702           |
|   310      62       5.0   5.00|(  3.72  1.79)|Maximum             |      |   .00   .00 |  900 900           |
|   310      62       5.0   5.00|(  3.72  1.79)|Maximum             |      |   .00   .00 | 1064 1064          |
|   304      62       4.9   4.92|   1.21   .32 | 2.54  1.5  1.44   .7| 1.12 |   .15   .12 |   79 79            |
|   304      62       4.9   4.92|   1.21   .32 | 2.55  1.5  1.54   .8| 1.12 |   .14   .12 |  138 138           |
|   304      62       4.9   4.92|   1.21   .32 | 2.52  1.5  1.21   .5| 1.13 |   .18   .12 |  572 572           |
|   304      62       4.9   4.92|   1.21   .32 | 2.61  1.6  2.71  1.7| 1.09 |   .03   .12 |  697 697           |
|   304      62       4.9   4.92|   1.21   .32 | 2.44  1.5   .78   .0| 1.15 |   .25   .12 |  720 720           |
|   304      62       4.9   4.92|   1.21   .32 | 2.58  1.6  1.86  1.1| 1.11 |   .11   .12 |  730 730           |
|   303      62       4.9   4.91|   1.12   .29 | 2.39  1.5  3.52  2.2| 1.07 |  -.04   .14 |  999 999           |
|   302      62       4.9   4.90|   1.04   .26 | 2.30  1.5  2.79  1.8| 1.09 |   .03   .15 |  878 878           |
|   300      62       4.8   4.87|    .93   .23 | 2.42  1.7  2.31  1.5| 1.16 |   .13   .17 | 1100 1100          |
|   298      62       4.8   4.85|    .84   .20 | 1.97  1.4  1.63   .9| 1.11 |   .13   .19 |  262 262           |
|   298      62       4.8   4.85|    .84   .20 | 1.87  1.3   .90   .0| 1.15 |   .24   .19 |  280 280           |
|   298      62       4.8   4.85|    .84   .20 | 1.92  1.3  1.11   .3| 1.13 |   .20   .19 |  321 321           |
|   298      62       4.8   4.85|    .84   .20 | 1.92  1.3  1.07   .3| 1.14 |   .20   .19 |  336 336           |
|     .       .        .     .|      .    . |   .    .    .    .|   . |     .    . |    .   .           |
|     .       .        .     .|      .    . |   .    .    .    .|   . |     .    . |    .   .           |
|     .       .        .     .|      .    . |   .    .    .    .|   . |     .    . |    .   .           |
|    87      62       1.4   1.10|   -.70   .08 |  .44 -4.2   .37 -2.6|  .67 |   .75   .53 |  901 901           |
|    86      62       1.4   1.08|   -.71   .08 |  .71 -1.8   .97   .0|  .45 |   .52   .53 |  990 990           |
|    84      62       1.4   1.04|   -.72   .08 |  .70 -1.8   .95   .0|  .47 |   .50   .53 |  113 113           |
|    82      62       1.3   1.01|   -.73   .08 |  .53 -3.2   .44 -2.1|  .53 |   .66   .53 |  846 846           |
|    80      62       1.3    .97|   -.75   .08 |  .72 -1.7   .70  -.9|  .49 |   .53   .52 | 1060 1060          |
|-------------------------------+-------------+-------------------+------+-------------+--------------------|
|  Total   Total   Obsvd  Fair-M|        Model | Infit      Outfit   |Estim.| Correlation |                    |
|  Score   Count  Average Avrage|Measure  S.E. | MnSq ZStd  MnSq ZStd|Discrm| PtMea PtExp | Num  Studenten     |
|-------------------------------+-------------+-------------------+------+-------------+--------------------|
|  215.6    62.0     3.5   3.65|    .02   .10 | 1.09   .1   .97  -.2|      |   .46       | Mean (Count: 1118) |
|   55.9      .0      .9   1.02|    .45   .13 |  .30  1.3   .33  1.0|      |   .15       | S.D. (Population)  |
|   55.9      .0      .9   1.02|    .45   .13 |  .30  1.3   .33  1.0|      |   .15       | S.D. (Sample)      |
+-------------------------------------------------------------------------------------------------------------+
   With extremes, Model, Populn: RMSE .16  Adj (True) S.D. .42  Separation 2.63  Strata 3.83  Reliability .87
   With extremes, Model, Sample: RMSE .16  Adj (True) S.D. .42  Separation 2.63  Strata 3.84  Reliability .87
Without extremes, Model, Populn: RMSE .09  Adj (True) S.D. .35  Separation 3.74  Strata 5.32  Reliability .93
Without extremes, Model, Sample: RMSE .09  Adj (True) S.D. .35  Separation 3.74  Strata 5.32  Reliability .93
With extremes, Model, Fixed (all same) chi-square: 15548.0  d.f.: 1117  significance (probability): .00
With extremes, Model,  Random (normal) chi-square: 684.6  d.f.: 1116  significance (probability): 1.00
-------------------------------------------------------------------------------------------------------------
```

Abb. 3.10 Table 7.2.1: Measurement Report für 'Studenten'

und Rasch-Thurstone ausgegeben, wobei in den Graphiken die Tresholds nach Rasch-Andrich zu finden sind.

Die zu Beginn dargestellte CPC zeigt die Wahrscheinlichkeit, dass eine Person mit einer bestimmten Fähigkeit eine bestimmte Anzahl von Antwortmöglichkeiten richtig markiert. Personen, die einen höheren Personenparameter aufweisen, findet man rechts auf der Graphik, da sie eine höhere Wahrscheinlichkeit aufweisen, die Fragen richtig zu beantworten. Der Übergang von einer Antwortkategorie zur nächsten ist der Treshold. Nachdem in unserem Beispiel allerdings die beiden äußeren Kategorien dominieren, findet man das auch in der Graphik wieder und man sieht den Übergang von Kategorie 0 auf 5; die anderen Kategorien verschwinden ziemlich am unteren Ende der X-Achse. Wie bereits erwähnt, gibt es auch die Möglichkeit die Graphiken extra aufzurufen. Man hat so die Möglichkeit, eine farbige Graphik zu bekommen, die man auch noch anpassen kann und in welche man zusätzlich die empirische CPC integrieren kann.

Anschließend wird die Item- Charakteristik Kurve (ICC) dargestellt, die die Beziehung zwischen der Fähigkeit einer Personen und der Wahrscheinlichkeit eine Aufgabe zu lösen, zeigt. Wie zuvor kann die modellierte und die empirische Kurve über den Menüpunkt 'Graphs' in einem Diagramm dargestellt werden, wobei auch noch das Konfidenzintervall ersichtlich ist.

3.6 Diskussion

Die Qualitätssicherung der Leistungsbeurteilungen sind mittlerweile ein fester Bestandteil jeder Universität und das MFRM erlaubt es, strenge und milde 'Beurteiler' bzw. Auswertungsschemen zu erfassen, sowie Aufschluss über die Fähigkeiten der beurteilen Personen und die Schwierigkeit der Fragen zu geben. Diese Aufgabe kann mit dem von Linacre (1989) entwickelten Programm FACETS durchgeführt werden, wobei die Messung der Beurteilerstrenge, der Personenfähigkeit und

Table 7.3.1 Fragen Measurement Report (arranged by mN).

Total Score	Total Count	Obsvd Average	Fair-M Avrage	Measure	Model S.E.	Infit MnSq	Infit ZStd	Outfit MnSq	Outfit ZStd	Estim. Discrm	Correlation PtMea	PtExp	Nu	Fragen
5249	2236	2.3	2.29	.41	.01	.82	-7.2	.81	-4.5	.76	.59	.57	4	Positionierung
5256	2236	2.3	2.30	.41	.01	.88	-4.6	.92	-1.8	.73	.57	.57	3	Segmentierung
6287	2236	2.8	3.02	.26	.01	.83	-6.8	.76	-5.8	.90	.60	.56	2	Einkaufslabor
6292	2236	2.8	3.02	.26	.01	1.02	.9	1.09	1.9	.90	.54	.56	15	Distributionsstufen
6332	2236	2.8	3.05	.25	.01	.86	-5.5	.84	-3.7	.90	.59	.56	17	Break-Even-Analyse
6789	2236	3.0	3.34	.18	.01	1.05	1.7	1.11	2.3	.86	.52	.54	31	Marktdurchdringung
6811	2236	3.0	3.35	.18	.01	.89	-4.1	.83	-3.7	.95	.57	.54	29	BCG-Portfolio
6975	2236	3.1	3.46	.16	.01	1.07	2.5	1.07	1.4	.89	.50	.54	24	Maslow
6992	2236	3.1	3.47	.15	.01	1.14	4.9	1.15	2.8	.86	.48	.54	9	Partial-/Totalm.
7042	2236	3.1	3.50	.15	.01	.88	-4.4	.80	-4.4	1.00	.58	.54	10	Verbundsysteme
7056	2236	3.1	3.50	.14	.01	.84	-6.0	.76	-5.1	1.00	.58	.54	6	Zeitl. Abstimmung
7075	2236	3.2	3.52	.14	.01	.88	-4.5	.82	-3.9	.98	.57	.54	7	Adoptionsprozess
7282	2236	3.2	3.64	.11	.01	.92	-2.7	.85	-2.9	.99	.55	.53	11	Kennzahlen
7475	2236	3.3	3.74	.08	.01	.94	-2.1	.92	-1.5	1.01	.54	.52	1	Branchenattraktivität
7709	2236	3.4	3.87	.04	.01	1.04	1.3	.97	-.5	.98	.50	.51	14	Datenerhebung
7794	2236	3.5	3.91	.02	.01	1.17	5.3	1.16	2.7	.97	.47	.51	8	Marktziele
7799	2236	3.5	3.91	.02	.01	.88	-4.0	.78	-4.1	1.07	.55	.51	28	Differenzierung
7832	2236	3.5	3.93	.02	.01	.97	-1.1	.89	-2.0	1.03	.52	.51	16	Wirtsch.-Aanlayse
7969	2236	3.6	4.00	-.01	.01	1.10	3.1	1.16	2.5	1.00	.48	.50	21	Werbestil
7970	2236	3.6	4.00	-.01	.01	1.26	8.0	1.35	5.4	.93	.43	.50	26	Kennzahlen/Interpr.
8469	2236	3.8	4.22	-.10	.01	1.23	6.4	1.20	2.9	1.00	.42	.48	23	Verkaufsförderung
8491	2236	3.8	4.23	-.10	.01	.96	-1.2	.84	-2.6	1.14	.52	.48	30	SWOT-Analyse
8621	2236	3.8	4.28	-.13	.01	1.29	7.8	1.32	4.4	1.01	.37	.47	13	Marketing-Aufgaben
8849	2236	4.0	4.36	-.18	.01	1.01	.2	.89	-1.6	1.12	.48	.45	18	Skimming Strategie
8978	2236	4.0	4.41	-.20	.01	1.00	.1	1.03	.4	1.14	.45	.45	22	Konfidenzintervall
9041	2236	4.0	4.43	-.22	.02	1.05	1.3	.98	-.2	1.13	.45	.44	12	Delphimethode
9389	2236	4.2	4.54	-.30	.02	1.19	4.5	1.04	.4	1.11	.41	.41	20	Involvement
9560	2236	4.3	4.59	-.35	.02	1.12	2.7	.85	-1.8	1.15	.42	.40	19	Geschl. Frage
9584	2236	4.3	4.60	-.35	.02	1.18	4.0	1.00	.0	1.13	.39	.39	27	Markoumfeld-Analyse
9626	2236	4.3	4.61	-.37	.02	1.17	3.6	.95	-.5	1.16	.40	.39	5	Mystery Shopping
10443	2236	4.7	4.81	-.69	.02	1.38	4.8	1.01	.1	1.16	.30	.27	25	Barter-Geschäft
Total Score	Total Count	Obsvd Average	Fair-M Avrage	Measure	Model S.E.	Infit MnSq	Infit ZStd	Outfit MnSq	Outfit ZStd	Estim. Discrm	Correlation PtMea	PtExp	Nu	Fragen
7775.4	2236.0	3.5	3.80	.00	.01	1.03	.3	.97	-.8		.49			Mean (Count: 31)
1268.2	.0	.6	.64	.24	.00	.15	4.4	.16	2.9		.08			S.D. (Population)
1289.2	.0	.6	.65	.25	.00	.15	4.5	.16	3.0		.08			S.D. (Sample)

Model, Populn: RMSE .01 Adj (True) S.D. .24 Separation 17.30 Strata 23.40 Reliability 1.00
Model, Sample: RMSE .01 Adj (True) S.D. .25 Separation 17.59 Strata 23.78 Reliability 1.00
Model, Fixed (all same) chi-square: 7674.1 d.f.: 30 significance (probability): .00
Model, Random (normal) chi-square: 29.9 d.f.: 29 significance (probability): .42

Abb. 3.11 Table 7.3.1: Measurement Report für 'Fragen'

der Kriterien- bzw. Aufgabenschwierigkeit auf einer gemeinsamen linearen Skala (Logitskala) im sogenannten Facettenraum dargestellt werden konnte. Das MFRM ist eine Erweiterung des polytomen Rasch-Modells, welches wie es Wright (1998) ausdrückt Folgendes zum Ziel hatte: 'I don't want to know which questions you answered. I want to know how much ... you know. I need to leap from what I know and don't want – to what I want but can't know. That's called inference.'

Die exemplarische Darstellung eines empirischen Beispiels (anhand von Multiple-Choice-Frage-Daten) in diesem Artikel zeigt die prinzipielle Anwendung des Programms, wobei das Programm auch weitaus komplexere Modelle schätzen kann. So können nicht nur weitere Facetten integriert werden, es ist auch möglich, die Interaktion von Facetten gesondert zu betrachten sowie für Teilbereiche der Daten unterschiedliche Modelle zu rechnen. Als ein Vorteil des MFRM im Allgemeinen, und von FACETS im Besonderen, kann sicherlich gesehen werden, dass durch die explizite Berücksichtigung von unterschiedlichen Facetten eine differenziertere Betrachtung von Personenparametern erreicht werden kann. So können, wenn bspw. unterschiedliche Beurteiler agieren, Aussagen getroffen werden, ob ein hoher Fähigkeitsparameter einer Person auf eine tatsächlich hohe Fähigkeit dieser Person zurückzuführen ist, oder (jedenfalls teilweise) durch die Milde des Beurteilers verursacht wird. Zusammenfassend kann gesagt werden, dass ein Rasch-Modell gilt, wenn eine Test-Performance vorrangig durch die Fähigkeit einer Person und die Schwierigkeit eines Items bestimmt wird und weitere Verzerrungen ausgeschlossen werden.

Ist ein Rasch-Modell gescheitert, hilft es immerhin noch insofern weiter, dass man Anhaltspunkte bekommt, ob das Modell z.B. an Personen- oder Itemhomogenität gescheitert ist. Generell ist ein fehlender Modell-Fit (wie im vorliegenden Beispiel) in empirischen Rasch-Analysen nichts ungewöhnliches, weil viele, teilweise widersprüchliche Kriterien, wie etwa Schnelligkeit oder Genauigkeit mit

```
-------------------------------------------------------------------------------------------------------------
Table 8.1  Category Statistics.

 Model = ?,?,?,R5
+-------------------------------------------------------------------------------------------------------------+
|      DATA            |  QUALITY CONTROL  |RASCH-ANDRICH|  EXPECTATION  |  MOST  |  RASCH-  | Cat| Obsd-Expd|
| Category Counts  Cum.| Avge  Exp.  OUTFIT| Thresholds  |  Measure at   |PROBABLE| THURSTONE|PEAK|Diagnostic|
|Score   Used   %    % | Meas  Meas   MnSq |Measure  S.E.|Category  -0.5 |  from  |Thresholds|Prob| Residual |
|-------------------+-------------------+-------------+---------------+--------+----------+----+----------|
|  0    15373  22%  22%|  -.18   -.15   .9 |           |(  -.85)       |   low  |   low    |100%|       .8 |
|  1     1192   2%  24%|   .19   -.03  1.8 |  2.47    .01|   -.39    -.62|        |   -.15   |  4%|          |
|  2     2625   4%  28%|   .18*   .10   .9 |  -.76    .01|   -.13    -.25|        |   -.11   |  7%|          |
|  3     4989   7%  35%|   .30    .24   .8 |  -.47    .01|    .08    -.02|        |   -.05   | 10%|          |
|  4     6057   9%  44%|   .44    .40   .6 |   .13    .01|    .36     .20|        |    .05   | 11%|          |
|  5    38708  56% 100%|   .57    .58  1.2 | -1.37    .01|(   .92)    .61|    .00 |    .17   |100%|      -.6 |
+-------------------------------------------------------------------------------------------------------------+

Scale structure

Measr:-1.0                              0.0                              1.0
       +                                 +                                +
  Mode:<0----------------(^)-------------05-----------------(^)------------5>

Median:<0----------------(^)--------01223-^34-^45-----------(^)------------5>

  Mean:<0---(^)-----01-------^---12---^-23---^--34----^-------45--------(^)-5>
       +                                 +                                +
Measr:-1.0                              0.0                              1.0

Probability Curves
      -1.0                              0.0                              1.0
       ++--------------------------------+---------------------------------++
     1 |                                                                    |
       |                                                                    |
       |000000000                                                           |
       |         00000                                              5555555|
       |              000                                    555555        |
     P |                 000                              5555              |
     r |                    00                         555                  |
     o |                      000                   555                     |
     b |                         00               55                       |
     a |                           0            555                         |
     b |                            00        55                            |
     i |                              00     55                             |
     l |                                00 55                               |
     i |                                 5*0                                |
     t |                               55   00                              |
     y |                              55       00                           |
       |                           555           00                         |
       |                       555                000                       |
       |                    5555 3333333**********************44444444444444|
       |11**********************************222222222222********33333333333|
     0 |**********4444444                    1111111111111111***************|
       ++--------------------------------+---------------------------------++
      -1.0                              0.0                              1.0

Expected Score Ogive (Model ICC)
      -1.0                              0.0                              1.0
       ++--------------------------------+---------------------------------++
     5 |                                                                    |
       |                                                          5555555555|
       |                                                    445555          |
       |                                               44444                |
     4 |                                            444                     |
       |                                         44                         |
       |                                      344                           |
       |                                    33                              |
     3 |                                  33                                |
       |                                 3                                  |
       |                               23                                   |
       |                             22                                     |
     2 |                           22                                       |
       |                        22                                          |
       |                      12                                            |
       |                    11                                              |
     1 |                  111                                               |
       |              1111                                                  |
       |         000011                                                     |
       |0000000000                                                          |
     0 |                                                                    |
       ++--------------------------------+---------------------------------++
      -1.0                              0.0                              1.0
```

Abb. 3.12 Table 8.1 Category Statistics

einfließen. Unerwartet hohe Fehlerwerte helfen Anomalien aufzudecken. Wenn gewünscht, können Ausreißer von Analysen ausgeschlossen werden, um den Fit zu verbessern (Linacre, 1994b). Als ein Nachteil des Programms ist mit Sicherheit zu bemerken, dass als Schätzer automatisch uML verwendet wird, und dieser auch nicht verändert werden kann. Besser wäre cML, da dieser Schätzer keine Verteilungsannahmen voraussetzt.

Kapitel 4
WinMiRa 2001

Karl Ledermüller, Thomas Peschta und Wolfgang Ziniel

Zusammenfassung In diesem Kapitel wollen wir einen anwendungsorientierten Einblick in das Programm WinMiRa geben, mit dem es unter anderem möglich ist Latent Class Analysen, Rasch Modelle und Mixed Rasch Modelle zu berechnen. Nach einer theoretischen Einführung wird das Programm WinMiRa vorgestellt. Der Fokus dieses Kapitels liegt auf einer beispielhaften Berechnung und Interpretation einer Latent Class Analyse und eines Mixed Rasch Modells mit Hilfe von WinMiRa.

4.1 Theorie Raschmodelle

4.1.1 Messtheoretische Fundierung

RASCH-Modelle folgen dem Paradigma der probabilistischen Testtheorie. Während die Klassische Testtheorie den Summenwert einer Person als einen intervallskalierten Messwert annimmt, stellt sich die Probabilistische Testtheorie die Frage, ob dies überhaupt zulässig ist, da Rohwerte grundsätzlich nur ordinales Skalenniveau haben. Deshalb erfolgt in einem ersten Schritt eine Prüfung ob es überhaupt zulässig ist, Items zu einem Summenwert als Maß der Personenfähigkeiten zusammenzufassen. Durch die Schätzungen der Personenfähigkeiten werden Personenparameter bestimmt, welche die Ausprägung einer Person auf einer latenten Variablen repräsentieren. Im Gegensatz zur Klassischen Testtheorie werden somit konkrete Verhaltensvorhersagen möglich, da berechnet werden kann mit welcher Wahrscheinlichkeit eine bestimmte Person bei Kenntnis von Itemschwierigkeit und Personenfähigkeit ein Item richtig beantwortet.

Die Probabilistische Testtheorie betrachtet dazu Antwortmuster und deren statistische Verteilungen da angenommen wird, dass diese von der Ausprägung der Eigenschaft abhängen (von Davier, 1997). Die Klassische Testtheorie hingegen fokussiert Rohwertvarianzen, Korrelationen und Kovarianzen der Itemantworten. Die Probabilistische Testtheorie versteht Antworten auf Items als manifeste Symptome einer latenten Variable. Wenn die Antwort von mehr als einer latenten Fähigkeit abhängt (z.B.: von einem Rate- oder Trennschärfeparameter) spricht man von Item-Response-Modellen. Im Gegensatz dazu geht der klassische Ansatz davon aus, dass eine Eigenschaft nur dann genau gemessen wird, wenn sie theoretisch unendlich oft gemessen werden könnte. Die Schätzung des Messfehlers steht also im Mittelpunkt der klassischen Betrachtungsweise.

WU Wien, Institute for Finance, Banking and Insurance • karl.ledermueller@wu.ac.at
thomas.peschta@wu.ac.at
WU Wien, Institut für Marketing-Management • wolfgang.ziniel@wu.ac.at

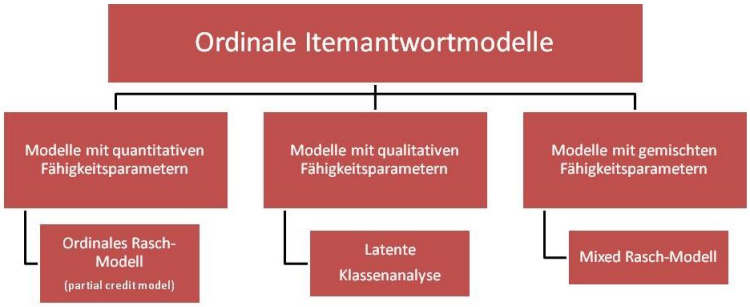


Abb. 4.1 Überblick über die Rasch Modelle

4.1.2 Wichtige Vorteile Probabilistischer Testmodelle

In Abgrenzung zur klassischen Testtheorie, die Zahlen einer Ratingskala wie metrische Variablen behandelt, wird bei probabilistischen Modellen wird die Wahrscheinlichkeit einer Antwort modelliert.

Ein weiterer Vorteil liegt darin, dass spezifische Objektivität im Kontext von psychologischen Tests erreicht werden kann. Nur dann sind Vergleiche zwischen den untersuchten Personen invariant über die verwendeten Items und Maße. Des Weiteren bleiben Vergleiche zwischen Items invariant über die spezifischen Personen an denen sie kalibriert werden.

4.1.3 Erweiterungen des klassischen Rasch Modells

4.1.3.1 Latent Class Modell

Latent Class Modelle setzen beobachtete multivariate und diskrete Variablen in Zusammenhang mit latenten Variablen. Dabei können latente Personenvariablen im Rahmen von Itemfunktionen aufgrund wahrscheinlicher Antwortmuster berücksichtigt werden. Bei nominalen Personenvariablen werden die Kategorien nicht so angeordnet, dass alle Itemfunktionen monoton steigen, wohingegen bei ordinalen Personenvariablen alle Itemfunktionen monoton steigen. So kann etwa die Klassenbildung über die Ausprägungen der Personenvariable z.B.: Klasse 1: „fähige Personen“ und Klasse 2: „nicht fähige Personen“ erfolgen.

4.1.3.2 Mixed-Rasch-Modell

Mixed Rasch Modelle gehen von der Existenz unterschiedlicher Klassen von Personen aus, für die das Rasch-Modell gilt. Es wird einerseits quantifiziert (also die Eigenschaftsausprägungen einer Person bestimmt) und klassifiziert (die Klassenzugehörigkeit ermittelt). Die Itemparameter werden so bestimmt, dass sie sich innerhalb der Klassen maximal unterscheiden. Dabei werden maximal unterschiedliche Antwortmuster gesucht und anhand dieser in Klassen eingeteilt. Passen Mixed-Rasch-Modelle zu den Daten kann auf eine unterschiedliche Lösungsstrategie / unterschiedliche Eigenschaften der Klassenmitglieder bei der Itembearbeitung geschlossen werden bzw. unterschiedliche Persönlichkeitstypen identifiziert werden.

4.1.4 Annahmen über threshold parameter im polytomen (Mixed) Rasch Modell

Die Modellbeschreibungen orientieren sich an den Ausführungen von Bühner (2006); von Davier (2001).

4.1.4.1 Ordinal (Partial Credit) Model

Partial Credit Modelle gehen davon aus, dass die Schwellen geordnet sind, da nicht aufsteigende Schwellenparameter negativ auf den Modellfit auswirken. Bei Anwendung des ordinalen Rasch-Modells muss dies durch Inspektion der Schwellenparameter geprüft werden. Um eine bestimmte Eigenschaftsausprägung zu erreichen muss die Person aufeinanderfolgende Schwellen erfolgreich überschreiten.(Siehe Abbildung 4.2). Dieses Modell wurde ursprünglich für Items mit teilrichtigen Lösungen (partial-credit) entwickelt. Die Ordnung der Schwellen muss nicht zwingend gegeben sein, da selten Vermutungen darüber bestehen welche Teilaufgabe schwierigerer/leichter sind. Das Partial Credit Model hat pro Item i und s Schwelle einen Parameter und zwar τ_{ik} (Rost, 2004, S. 109f). Die Schwellen werden in diesem Modell ohne Restriktionen geschätzt. Dabei müssen nicht alle Items dieselbe Anzahl von Kategorien haben, wie dies beim Rating Scale Model, Equidistance Model bzw. Dispersions Model der Fall ist.

Die allgemeine Notation des ordinalen Klassenmodells lautet:

$$p(X_{vi} = x) = \pi \frac{exp(x\Theta_i - \sum_{s=1}^{x} \tau_{is})}{\sum_{s=0}^{m} exp(s\Theta_i - \sum_{t=1}^{s} \tau_{it})}$$

Das Modell gibt die Wahrscheinlichkeit einer Person v bei einem Item i die Kategorie x zu wählen wieder. Diese Schwellenwahrscheinlichkeit bezeichnet den Anteil der höheren an der niedrigeren Kategorienwahrscheinlichkeit (der relative Anteil der höheren Kategorienwahrscheinlichkeit an beiden Kategorienwahrscheinlichkeiten). Die Schwellenwahrscheinlichkeit gibt also den Anteil der Wahrscheinlichkeit eine Kategorie „sehr häufig“ an der Wahrscheinlichkeit eine Kategorie „öfters“ zu wählen. s spezifiziert die Nummer der Schwelle, während m für die Anzahl der Schwellen steht. Der Personenparameter Θ gibt die Fähigkeitsausprägung der Person beim Item i an. Θ_i repräsentiert schließlich den Schwellenparameter des dahinterliegenden Modells. Dieser ist interpretierbar als die Schwierigkeit eine bestimmte Schwelle zu überschreiten. Wenn der Schwellenparameter einer Person dem Personenparameter derselben entspricht, ist die Wahl der beiden benachbarten Aussagenkategorien gleich wahrscheinlich. Die Person liegt im latenten Kontinuum genau zwischen den beiden Aussagekategorien (Bühner, 2006, S.330ff.).

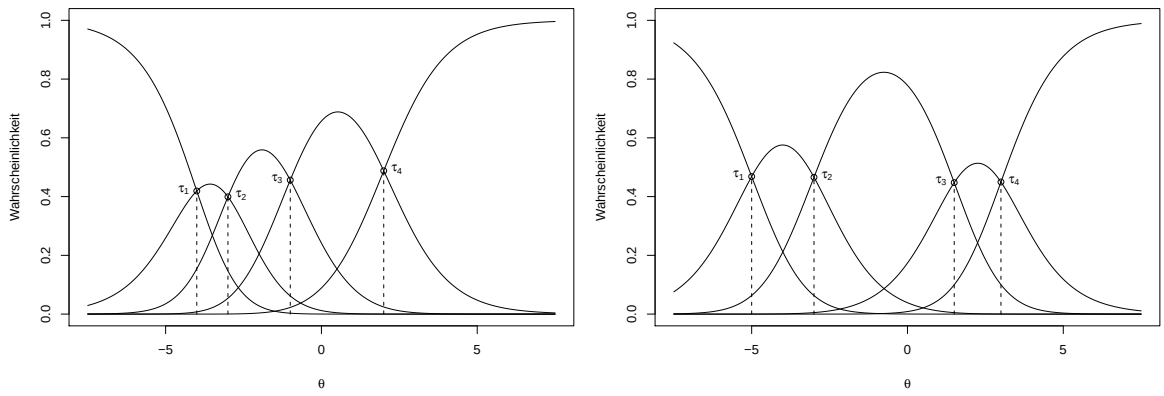


Abb. 4.2 Category Characteristic Curves Partial Credit Model

4.1.4.2 Rating Scale Model

Die grundlegende Überlegung bei Modellen für Ratingskalen liegen darin, einen Schwierigkeitsparameter für jedes Item zu bestimmen und die Schwellenparameter über alle Items zu modellieren. Mögliche Annahmen wären die Äquidistanz (Schwellen der Ratingskala haben gleiche Abstände) und jene, dass die Schwellen unterschiedliche Abstände haben, diese Abstände jedoch für alle Items gleich sind. Aufgrund unterschiedlicher Kombinationsmöglichkeiten dieser Annahmen entstehen verschiedene Modelle, die Antwortprozesse bei Verwendung von Ratingskalen formalisieren (Rost, 2004, S. 215). Rating Scale Modelle nehmen an, dass die Abstände zwischen zwei aufeinander folgenden Schwellen über alle Items gleich sind. Jedoch variiert die Lokalisation der Schwellen von Item zu Item, je nach Schwere der Items. Es leitet sich aus der Verwendung desselben Antwortformats für alle Items eines Tests ab. Man stelle sich beispielweise als Antwortformat die österreichischen Schulnoten vor. Falls alle Items eines Fragebogens dieses Antwortformat verwenden, muss der Abstand der Schwellenparameter zwischen den Kategorien nicht genügend/genügend und genügend/befriedigend nicht derselbe sein wie zwischen befriedigend/gut und gut/sehr gut. Über alle Items sollten zwei aufeinanderfolgende Schwellenparameter aber denselben Abstand aufweisen (Effekt der Kategorienbenennung). Items unterscheiden sich nur anhand ihrer Schwierigkeit. Unterschiedliche Itemparameter werden durch die Fragenformulierung erreicht (Siehe Abb. 4.3). Ratingskalenmodelle fußen auf der Annahme dass die Schwellenabstände aller Items gleich sind. Die Schwellenabstände sind keine Eigenschaft der Items mehr, sondern eine Eigenschaft des Antwortformats. Das Modell enthält statt der doppelt indizierten Parameter t_{ix} nur mehr das einfach indizierte t_x, welches für alle Items gilt. Dadurch verringert sich die Anzahl der Parameter da nicht $k \cdot (m-1)$ sondern nur $m-1$ Parameter benötigt werden. Das Modell eignet sich gut bei Ratingskalen mit Items die jeweils die gleiche Anzahl der Antwortkategorien haben.

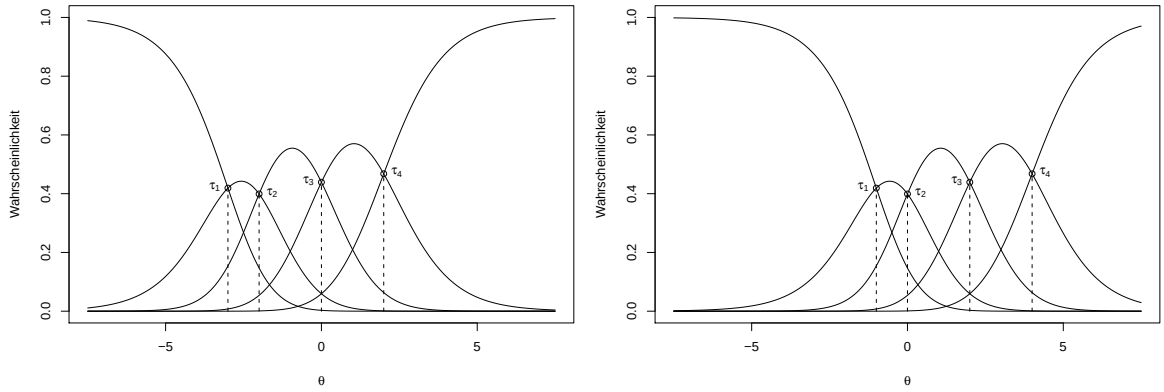


Abb. 4.3 Category Characteristic Curves Rating Scale Model

4.1.4.3 Equidistance Model

Das Equidistance Modell nimmt an, dass die Abstände zwischen zwei aufeinander folgenden Schwellen für jedes und über alle Items gleich sind. Dieses Modell liefert einen Hinweis darauf, dass eine Skala bzw. ein Item wirklich intervallskaliert ist. Es ist aber möglich dass der konstante Abstand für jedes Item unterschiedlich groß ist. Anstatt der Schwellenparameter t_{ix} enthält das Modell den Distanzparameter δ_i als zusätzlichen Itemparameter. Durch den Koeffizienten dieses Parameters $((x-(m+1)/2)$ wird gewährleistet, dass jede Schwelle durch ihren Abstand zum Mittelpunkt aller Schwellen definiert wird. Statt der $k-(m-1)$ unabhängigen Schwellenparameter beinhaltet das Modell nur k Distanzparameter. Siehe Abbildung 4.4.

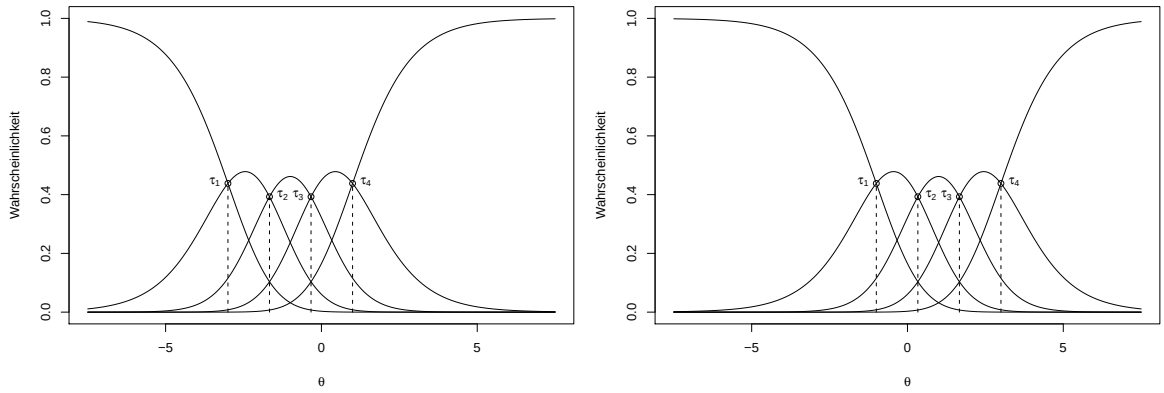


Abb. 4.4 Category Characteristic Curves Equidistance Model

4.1.4.4 Dispersion Model

Das Dispersionsmodell enthält sowohl Equidistance Parameter als auch Threshold Parameter. Letztere definieren die Distanzen für die Thresholds welche mittels der Equidistance Parameter erhöht bzw. vermindert werden. Die Schwellenabstände werden als eine Eigenschaft des Antwortformats gesehen und sind deshalb für alle Items gleichartig. Das Modell enthält item-unabhängige Schwellenparameter t_x. Um den Einfluss der Items auf die Antwortstreuung bzw. die Schwellenabstände zu berücksichtigen, wird ein Distanzparameter δ_i eingeführt. Mittels δ_i kann der Schwellenabstand bei jedem item um δ_i vermindert/erhöht werden. Je größer δ für ein item, desto kleiner die Streuung der Antworten über die Kategorien. Die Schwellenabstände werden anstatt durch $k \cdot (m-1)$ Parameter durch $(k-1) \cdot (m-1)$ Parameter festgelegt. In der nachfolgenden Darstellung ist die zweite Schwellenwertdistanz größer als die erste. Siehe Abbildung 4.5.

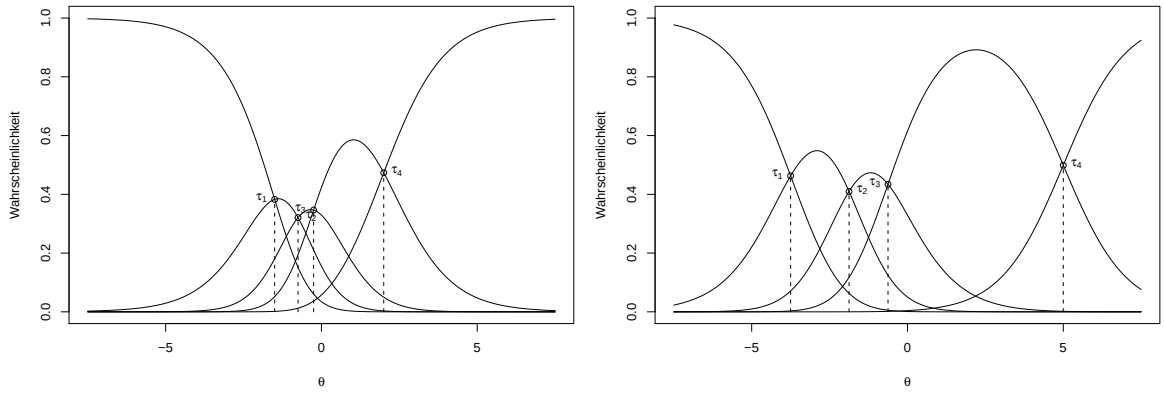


Abb. 4.5 Category Characteristic Curves Dispersion Model

4.2 Programmbeschreibung

4.2.1 unterstützte Modelle

Die Software unterstützt sowohl Rasch Modelle als auch die Latent Class Analysis. Die Mischform dieser beiden Analysen, i.e. Mixed Rasch Modelle ist ebenfalls möglich. Die Software verwendet das einfache Rasch Modell als Sonderform des Mixed-Rasch Modelles mit lediglich einer latenten

Klasse. Zusätzlich können Hybridmodelle gerechnet werden, bei denen für unterschiedliche Klassen unterschiedliche Modelle eingestellt und berechnet werden können.

4.2.2 unterstützte Skalenniveaus

WinMiRa unterstützt sowohl dichotome als auch polytome Daten. Bei polytomen Daten werden folgende vier verschiedene Rasch Modelle unterstützt, die in Kapitel 4.1.4 näher beschrieben wurden.

- Rating Scale Model
- Equidistance Model
- Dispersion Model
- Ordinal bzw. Partial Credit Model

4.2.3 Usability

4.2.3.1 Programminstallation

WinMiRa ist unter Windows out-of-the-box möglich, das Programm unterstützt jedoch nur dieses Betriebssystem. Eine Trial Version kann unter http://winmira.von-davier.de/wmira/index.html heruntergeladen werden (selbstextrahierendes .exe file.)

4.2.3.2 verfügbare Beschreibungen

Es finden sich zur Software mehrere hilfreiche, gut auskommentierte Quellen, die den Einstieg erleichtern. Als Referenzen seien genannt:

- **Die WinMiRa Homepage:** http://winmira.von-davier.de/wmira/index.html Die Seite kann verwendet werden um wie oben angeführt eine Trial Version herunterzuladen und/oder eine Vollversion zu kaufen.
- **Das online WinMiRa Manual:** (von Davier, 2001) Das Manual beinhaltet eine umfassende Erklärung des Programms anhand von Beispielen.
- Das mit dem Programm mitgelieferte WinMiRa Kurzmanual: (Reimers, 2005)
- Ein weiteres Kurzmanual wird mit dem Programm mitgeliefert, ist im .html Format abrufbar und dient als Kurzmanual, welches den Programmablauf erörtert.
- Diverse Bücher verwenden in der Erklärung von Rasch Modellen WinMiRa wie beispielsweise:(Bühner, 2006)oder auch (Rost, 2004)

4.2.3.3 Handling

Das Programm ist aufgrund des Graphical User Interfaces (GUI) einfach zu bedienen. Der Menüaufbau ist intuitiv.

4.2.3.4 Schnittstellen

WinMiRa unterstützt zum Datenimport ASCII Tabellen mit der Dateiendung .dat. Die Datentabelle darf nach dem Freefield Format (Spaltentrennung mittels Leerzeichen) mittels eines vordefinierten Trennzeichens oder ohne Trennzeichen aufgebaut sein. Des weiteren werden SPSS Dateien unterstützt. Datenfiles können wieder als ASCII Tabellen und im SPSS Format exportiert werden. Das

Outputfile (als .OUT -ASCII) kann ebenso wie Plots (.WMF und .BMP) exportiert werden. Definitionsdateien (.ini) können sowohl gelesen als auch geschrieben werden. Sie beinhalten Informationen über programmspezifische (Bsp.: Pfad für Outputfiles) wie auswertungsspezifische (Bsp.: Itemlabels, Iterationsobergrenzen...) Einstellungen.

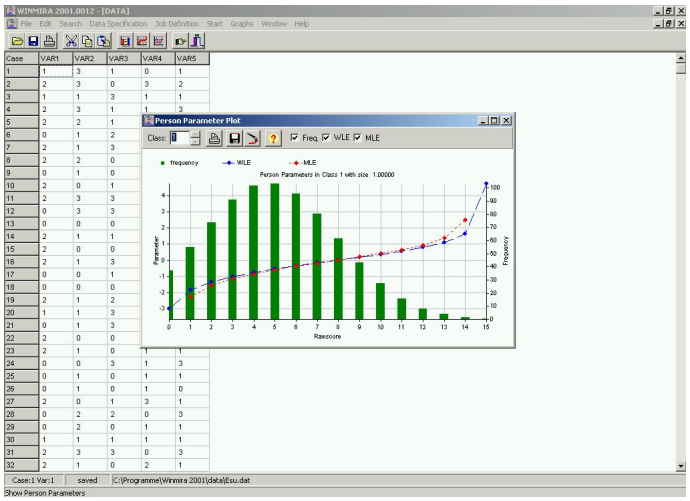


Abb. 4.6 Screenshot WinMiRa mit Person Parameter Plot (Datensatz: Esu.dat)

4.2.4 grundsätzliche Bedienung

Die generelle Vorgehensweise bei der Durchführung einer Analyse mit WinMiRa ist folgende:

- Laden des Datensatzes: <File> <Open> <Get Ascii Data oder Open Spss Data>
- optional: Durchführen von Recodings und Behandlung von Missing Values: <Data Specification> <Recodings bzw. <Missing Value>
- Variablen auswählen: <Data Specification> <Select Variables> Tipp: alle Variablen können mittels Doppelklick auf eine Variable hinzugefügt bzw. entfernt werden.
- Anzahl der Klassen definieren: <Job Definition> <N. of. Classes> Hinweis: Wird das klassische Rasch Modell gerechnet ist die Anzahl der Klassen auf 1 zu belassen. Das Mixed Modell (mindestens 2 Klassen) beinhaltet eine Kombination zwischen einer Latent Class Analyse und einem Rasch Modell und bietet somit die Möglichkeit, klassenspezifische Auswertungen durchzuführen.
- Modell auswählen: <Job Definition> <Select Model> - Als Standard wird das klassische Rasch Modell verwendet. Bei polytomen Daten kann je nach Annahme über die Lage der thresholds zwischen dem Rating Scale Model, dem Equidistance Model, dem Dispersion Model und dem Ordinal bzw. Partial Credit Model unterschieden werden.
- optional und fortgeschritten: Ändern der Parameter constraints: <Job Definition> <Parameter constraints> Ändern diverser Parametereinstellungen
- optional: Ändern diverser Parameter im Output Bericht <Job Definition> <Output Options>
- optional: Einstellungen zum Bootstrap Test <Job Definitions> <Bootstrap Options>
- Berechnung starten: <Start> <Start Job> bzw. F9; ein Outputfile wird ausgegeben.
- diverse Grafiken anzeigen: <Graphs> Hier ist das Anzeigen diverser Grafiken (Category Probability Plot F5 Abbildung: 4.7; Item Parameter Plot F6 Abbildung:4.8 und Person Parameter Plot F7 Abbildung: 4.6.

Das Outputfile ist ein ASCII Textfile und enthält diverse Summaries. Eine genaue Erklärung der Zusammensetzung und der Interpretation des Outputfiles findet sich unter Punkt 4.3.

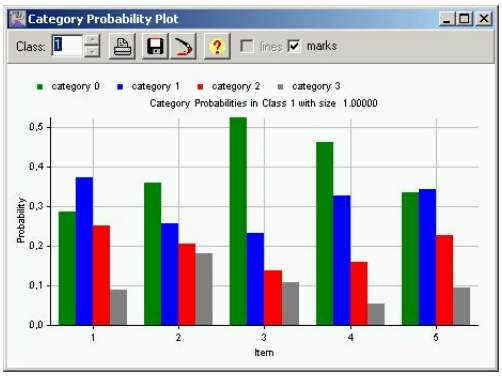


Abb. 4.7 Category Probability Plot (Datensatz: Esu.dat)

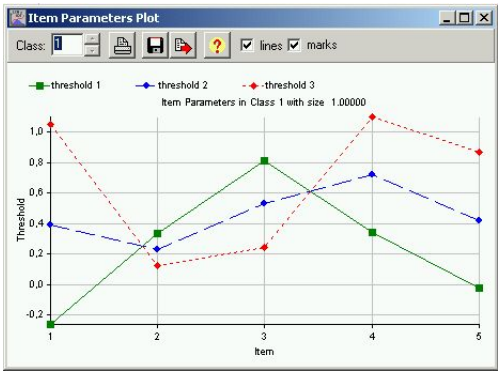


Abb. 4.8 Item Parameters Plot (Datensatz: Esu.dat)

4.3 Beispiele

4.3.1 Latent Class Analyse mit dichotomen Daten

Das Ziel der Latent Class Analyse ist es, das Vorliegen einer kategorialen latenten Personenvariable aufgrund wahrscheinlicher Antwortmuster der Befragten nachzuweisen. Die Analyse latenter Klassen basiert auf gewissen Grundannahmen, die wie folgt lauten: (Rost, 2004, S. 153ff)

- Die Lösungswahrscheinlichkeit eines Items ist für alle Personen innerhalb einer Klasse gleich groß.
- Jede Person kann nur einer Klasse angehören.
- Alle Items messen dieselbe Personenvariable.
- Die Lösungswahrscheinlichkeiten der Items sind stochastisch unabhängig voneinander.

Die Zuordnung einer Person zu einer Klasse aufgrund derer Antworten erfolgt dabei jedoch nicht „manifest“ sondern nur anhand einer gewissen Wahrscheinlichkeit (Rost, 2004, S. 156). Wie eine derartige Analyse mit WinMiRa möglich ist, wird nun anhand des Beispieldatensatzes „KFT“ der im Lieferumfang von WinMiRa enthalten ist (Dateiname: kft.dat) dargestellt. Der Datensatz ist ein Auszug aus einem kognitiven Fähigkeitstest und umfasst die Antworten von 300 Personen auf fünf Items eines Intelligenztests. Eine richtige Antwort einer Person wurde als „1“ codiert, eine falsche Antwort wurde mit „0“ codiert. Die kognitive Fähigkeit einer Person richtet sich nach der Anzahl der richtigen beziehungsweise falschen Antworten (Rost, 2004, S. 99). Im Rahmen dieses Beispiels wird nun versucht, die befragten Personen aufgrund ihres Antwortverhaltens in „kognitiv fähige“ und „nicht kognitiv fähige“ Personen zu unterteilen. Die Kategorien „kognitiv fähig“ und „nicht kognitiv fähig“ können dabei als Kategorien der latenten Variable „kognitive Fähigkeit“ angesehen werden. Nach dem Öffnen des Datensatzes „KFT.dat“ in WinMiRa ist zunächst die Auswertung zu spezifizieren:

- Auswahl der Variablen „Var 1“ bis „Var 5“ unter <Data Specification>

- Definition der Anzahl der zu bildenden Klassen unter <Job Definition> <N. of. Classes>. In diesem Beispiel sollen zunächst zwei latente Klassen gebildet werden.
- Auswahl der Latent Class Analyse unter <Job Definition> <Select Model> <Latent Class Analysis> Durch den Befehl <Start> <Start Job> wird das gewählte Modell berechnet und die Ergebnisse der Analyse in Textform dargestellt.

Nach einer Zusammenfassung der Antworthäufigkeiten je untersuchter Variable werden die Schätzergebnisse der latenten Klassen dargestellt, auf die nun näher eingegangen wird (Abb. 4.9).

```
Final estimates in CLASS 1 of 2 with size  0.53791
==================================================



expected category frequencies and item scores:

   Item    |     Item's    | relative category
   label   | Score | Stdev |    frequencies
 ----------|-------|-------|    0    |    1
  VAR1     |  0.90 |  0.30 |  0.102  |  0.898
  VAR2     |  0.93 |  0.25 |  0.067  |  0.933
  VAR3     |  0.74 |  0.44 |  0.256  |  0.744
  VAR4     |  0.66 |  0.47 |  0.336  |  0.664
  VAR5     |  0.48 |  0.50 |  0.518  |  0.482

    Sum:   |  3.72

threshold parameters:  ordinal (partial credit) model

  item      |   item    |
  label     | location  | threshold parameters
------------|-----------
  VAR1      | -2.18001
  VAR2      | -2.63564
  VAR3      | -1.06849
  VAR4      | -0.68340
  VAR5      |  0.07036
```

Abb. 4.9 Schätzergebnisse der latenten Klassen(Klasse 1) - KFT.dat

Die erste Klasse umfasst rund 54% der befragten Personen. Der Spalte „relative category scores“ können die Antwortwahrscheinlichkeiten von Personen, die dieser Klasse zugeordnet wurden, entnommen werden. So liegt die Wahrscheinlichkeit, dass die Personen dieser Klasse das Item 1 (VAR1) richtig beantworten bei rund 90%. Bei Item 2 (VAR 2) liegt diese Lösungswahrscheinlichkeit bei rund 93%. Diesen Personen fällt die richtige Beantwortung dieser beiden Fragen also leicht. Hingegen liegt die Lösungswahrscheinlichkeit bei Item 5 (VAR 5) lediglich bei rund 48%, die Wahrscheinlichkeit diese Frage nicht richtig zu beantworten liegt vice versa bei 52%. Für diese Personen ist somit eine falsche Beantwortung dieser Frage wahrscheinlicher als die richtige Beantwortung dieser Frage. In der Tabelle sind „threshold parameters“ die Item-Parameter für diese latente Klasse dargestellt. Dabei würde ein Item, das eine Lösungswahrscheinlickeit von 50% aufweist, einen Item-Parameter von 0 aufweisen. Items mit einer höheren Lösungswahrscheinlichkeit weisen einen negativen Item-Parameter auf, Items mit einer geringeren Lösungswahrscheinlichkeit als 50% weisen einen positiven Item-Parameter auf. Die Itemparameter können dabei grundsätzlich zwischen $\pm\infty$ liegen, wobei zumeist Werte zwischen ± 4 zu erwarten sind. Somit sind die Fragen 1-4 (VAR 1-4) unterdurchschnittlich leicht für die Personen dieser Gruppe, da die Lösungswahrscheinlichkeit hier jeweils größer als 50% ist und somit negative Itemparameter ausgewiesen werden. Frage 5 ist hingegen, aufgrund der geringen

Lösungswahrscheinlichkeit von 48%, überdurchschnittlich schwierig für diese Personen, weshalb auch der Itemparameter in diesem Fall positiv ist. Die Personen dieser Klasse weisen somit eine hohe Wahrscheinlichkeit auf, zumindest vier der insgesamt fünf Fragen richtig beantworten zu können.

```
Final estimates in CLASS 2 of 2 with size  0.46209
==================================================


expected category frequencies and item scores:

   Item    |     Item's    | relative category
   label   | Score | Stdev |    frequencies
 ----------|-------|-------|    0    |    1
  VAR1     |  0.36 |  0.48 |  0.639  |  0.361
  VAR2     |  0.18 |  0.38 |  0.824  |  0.176
  VAR3     |  0.17 |  0.37 |  0.835  |  0.165
  VAR4     |  0.04 |  0.20 |  0.958  |  0.042
  VAR5     |  0.12 |  0.32 |  0.883  |  0.117

    Sum:   |  0.86

threshold parameters:  ordinal (partial credit) model

  item      |   item    |
  label     | location  | threshold parameters
------------|-----------
  VAR1      |  0.57192
  VAR2      |  1.54264
  VAR3      |  1.62070
  VAR4      |  3.13697
  VAR5      |  2.02595
```

Abb. 4.10 Schätzergebnisse der latenten Klassen (Klasse 2) - KFT.dat

Gemäß Abb. 4.10 umfasst die zweite Klasse rund 46% der befragten Personen. Der Spalte „relative category scores" können die Antwortwahrscheinlichkeiten von Personen, die dieser Klasse zugeordnet wurden, entnommen werden. Die Wahrscheinlichkeit, dass die Personen dieser Klasse das Item 1 (VAR1) richtig beantworten liegt bei lediglich 36%. Die Gegenwahrscheinlichkeit, das heißt die Wahrscheinlichkeit, dass diese Personen diese Frage nicht richtig beantworten können, liegt bei rund 64%. Die Lösungswahrscheinlichkeit von Frage 2 (VAR2) und Frage 3 (VAR3) ist ähnlich hoch und beträgt 17% beziehungsweise 18%. Das schwierigste Item für diese Personen ist demnach Item „VAR4" da es lediglich eine Lösungswahrscheinlichkeit von rund 4% aufweist. Die Frage 5 (VAR5) weist in dieser Gruppe eine Lösungswahrscheinlichkeit von 12% auf. Die Schwierigkeit der Items innerhalb dieser Gruppe können der Tabelle „threshold parameters" entnommen werden. Dabei fällt auf, dass alle Itemparameter größer als 0 sind, da die Lösungswahrscheinlichkeiten dieser Items geringer als 50% sind. In dieser Klasse befinden sich somit vorrangig Personen welche die gestellten fünf Fragen eher nicht richtig beantworten konnten.

Aus der Tabelle „statistics of expected class membership" (Abb. 4.11) geht hervor, dass aufgrund der wahrscheinlichen Antwortmuster der Befragten die beiden Klassen jeweils eine erwartete Größe von 50% aufweisen (Spalte „exp. Size"). Über die Treffsicherheit der Klassenzuordnung gibt die Spalte „mean prob." Auskunft. Demnach beträgt die Zuordnungswahrscheinlichkeit einer Person zur ersten Klasse aufgrund ihres Antwortmusters rund 97%. Die Wahrscheinlichkeit einer falschen Zuordnung beträgt bei dieser Klasse lediglich 3%. Die Zuordnungswahrscheinlichkeit einer Person zur zweiten Klasse beträgt 89%, die Wahrscheinlichkeit einer Falschzuordnung zu dieser Klasse aufgrund des Antwortmusters einer Person beträgt hier 11%. Diese Werte sind grundsätzlich als gut

```
statistics of expected class membership:

      | exp.  | mean  |
 class | size  | prob. |   1   |   2   |
 ______|_______|_______|______|______|
     1 | 0.500 | 0.966 | 0.966| 0.034|
     2 | 0.500 | 0.890 | 0.110| 0.890|

Goodness of fit statistics:

                                 estimated       saturated
                                   model           model

 Log-Likelihood          :        -850.55         -830.39
 Number of parameters    :             11              31
 geom. mean likelihood   :      0.56720616      0.57487927

Information Criteria:

   AIC-Index             :        1723.10         1722.79
   BIC-Index             :        1763.84         1837.60
   CAIC-Index            :        1774.84         1868.60

Power Divergence GoF statistics:

                                  emp. value     chi-square p-value

   Cressie Read            :          38.63     p=  0.0074
   Pearson Chisquare       :          38.71     p=  0.0072

================================================================

   Likelihood ratio        :          40.31     p=  0.0046
   Freeman-Tukey  Chi^2    :          45.60     p=  0.0009

   Degrees of freedom      :             20
```

Abb. 4.11 Fit Statistiken - KFT.dat

einzustufen (Bühner, 2006, S.381). Die angeführten Fit-Statistiken sind im Rahmen dieser Analyse nicht interpretierbar aber für Modellvergleiche relevant (Bühner, 2006, S. 381). Die Ergebnisse der Latent Class Analyse können auch in graphischer Form ausgegeben werden. Der „Category Probability Plot“ (Abb. 4.12) zeigt etwa die Antwortwahrscheinlichkeiten auf die jeweiligen Fragen getrennt nach der latenten Klasse.

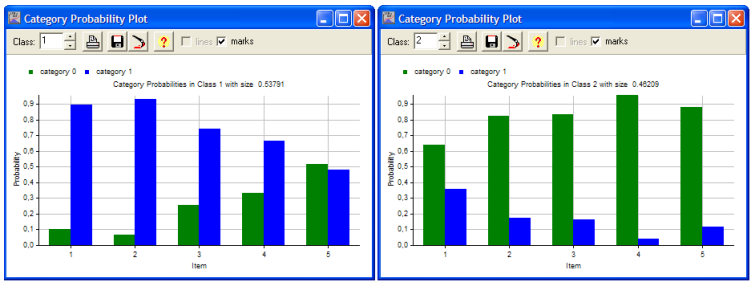


Abb. 4.12 Category Probability Plot - KFT.dat

Dabei erkennt man, dass die Personen der ersten Klasse bei allen Fragen eine deutlich höhere Lösungswahrscheinlichkeit haben als die Personen der zweiten Klasse. Dem „Person Parameter Plot“ (Abb. 4.13 kann im Rahmen der Latent Class Analyse lediglich die Häufigkeit der richtigen Antworten je latenter Klasse entnommen werden.

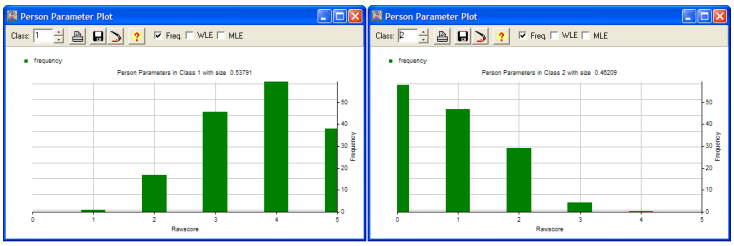


Abb. 4.13 Person Parameter Plot - KFT.dat

So gibt es in der ersten Klasse keine Personen, die auch nur eine der fünf Fragen falsch beantwortet hat, wohingegen in der zweiten Klasse mehr als 50% der Befragten keine der fünf Fragen richtig beantwortet haben. Hingegen hat ein Großteil der Personen der ersten Klasse vier der fünf Fragen richtig beantwortet, wohingegen der Anteil der Personen mit vier richtigen Antworten in der zweiten Klasse verschwindend gering ist.

Der „Item Parameters Plot“ (Abb. 4.14 fasst die Schwierigkeit der jeweiligen Fragen anhand der Itemparameter je nach Klasse zusammen.

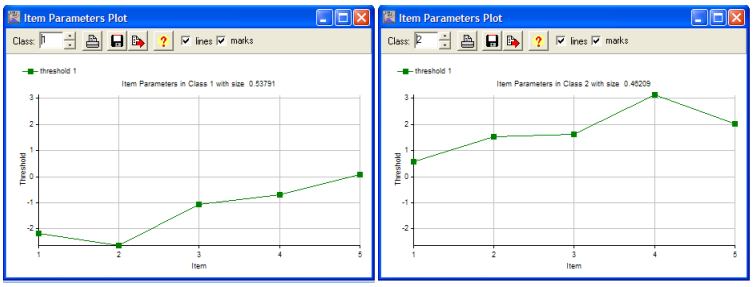


Abb. 4.14 Item Parameter Plot - KFT.dat

Dabei ist die Schwierigkeit der Fragen durch die Höhe des Thresholds definiert. Während für die Personen der ersten Klasse die richtige Beantwortung der zweiten Frage am leichtesten ist, ist die richtige Beantwortung dieser Frage für Personen, die der zweiten Klasse zugeordnet wurden, deutlich schwieriger. Abschließend kann somit festgehalten werden, dass es anhand der ermittelten Klassen möglich ist, zwischen Personen mit „hoher kognitiver Fähigkeit“ und „geringer kognitiver Fähigkeit“ zu unterscheiden. Demnach konnte das Vorliegen einer zumindest zweikategorialen latenten Personenvariable „kognitive Fähigkeit“ nachgewiesen werden. Ebenfalls kann untersucht werden, ob eine Lösung mit drei oder mehr Klassen besser ist als die soeben dargestellte Lösung mit zwei Klassen. Dazu ist die Zahl der Klassen unter <Job Definition> <N. of. Classes> je nach Bedarf einzustellen. Zur Entscheidung, welches Modell die beste Anpassungsgüte aufweist, können die Informationskriterien AIC, BIC oder CAIC herangezogen werden, die in der Tabelle „Information Criteria (Tab.: 4.3.1“ dargestellt werden.

Diese Kriterien geben keine absoluten Kriterien vor, wann ein Modell passt. Vielmehr wird ermittelt, welches Modell unter konkurrierenden und unterschiedlich komplexen Modelle als das geeignetste darstellt. Es sind also nur relative Vergleiche möglich. Je geringer der jeweilige Wert, desto besser "passt"das Modell auf die Daten, wenngleich auch alle Modelle weit vom saturierten Modell abweichen können. Der Vorteil dieser Kriterien liegt darin, dass die Komplexität des Modells und der Fit der Daten auf das Modell eingehen. Dem steht gegenüber, dass absolute Kriterienwerte nicht hinsichtlich

der Frage, ob ein Modell Daten gut beschreibt, beantwortet werden können. Für große Itemzahlen und kleinen Patternhäufigkeiten empfiehlt Bühner (2006, S.352) die BIC Kennzahl.

$$\text{Akaike Information Criterion (AIC)} = -2logL + 2n_p$$

$$\text{Bayes Information Criterion (BIC)} = -2logL + (logN) \cdot 2n_p$$

$$\text{Consistent AIC (CAIC)} = -2logL + (logN) \cdot n_p + n_p$$

- n_p: Anzahl der Parameter des entsprechenden Modells
- N: Stichprobengröße
- L: Likelihood

Die folgende Tabelle fasst diese Informationskriterien für Lösungen mit zwei, drei und vier Klassen zusammen:

	AIC	BIC	CAIC
2 Klassen	1.723,10	1.763,84	1.774,84
3 Klassen	1.709,24	1.772,20	1.789,20
4 Klassen	1.715,31	1.800,50	1.823,50

Tabelle 4.1 Information Criteria - KFT.dat

Demnach weist die Lösung mit zwei Klassen die geringsten BIC- und CAIC-Werte auf, weshalb diese Lösung zu favorisieren ist. Zwar zeigt diese Lösung das höchste AIC, was jedoch nicht verwundert, da dieses Maß keinen Strafterm für die Anzahl der im Modell berücksichtigten Parameter beinhaltet.

4.3.2 Mixed Rasch Modell mit WinMiRa

Bei Mixed Rasch Modellen werden sowohl die Eigenschaftsausprägungen einer Person und deren Klassenzugehörigkeit ermittelt. Somit ermöglicht diese Analyse, dass sich die Lösungswahrscheinlichkeit der Personen innerhalb einer Klasse graduell voneinander unterscheiden können, wohingegen im Rahmen der Latent Class Analyse unterstellt wurde, dass sich die Lösungswahrscheinlichkeiten innerhalb einer Klasse nicht unterscheiden dürfen. Um dies zu verdeutlichen, wird nun ein Mixed Rasch Modell anhand des zuvor beschriebenen KFT-Datensatzes berechnet. Um ein solches Modell in WinMiRa zu berechnen, sind folgende Einstellungen in WinMiRa vorzunehmen:

- Öffnen des Datensatzes „kft.dat“
- Auswahl der zu berücksichtigenden Variablen „Var 1“ bis „Var 5“ unter <Data Specification>
- Definition der Anzahl der zu bildenden Klassen unter <Job Definition> <N. of. Classes>. In diesem Beispiel sollen zwei latente Klassen gebildet werden.
- Auswahl der Latent Class Analyse unter <Job Definition> <Select Model> <Mixed Rasch Modell>

Durch den Befehl <Start> <Start Job> wird das gewählte Modell berechnet, die Ergebnisse der Analyse werden ebenfalls in Form eines Textberichtes dargestellt. Nach einer Zusammenfassung der Antworthäufigkeiten je untersuchter Variable werden die Schätzergebnisse der latenten Klassen dargestellt.

Eine Gegenüberstellung der Itemscores der beiden Klassen (KFT.dat) in (Tab. 4.3.2) zeigt eine deutlich höhere Lösungswahrscheinlichkeit der Variablen 2-5. Somit handelt es sich in dieser Klasse um „fähigere“ ProbandInnen.

Gemäß Abbildung 4.15 werden der ersten Klasse rund 59% der Befragten zugeordnet. Aus der angeführten Tabelle geht die erwartete Häufigkeit von Personen mit der jeweiligen Anzahl an richtigen Antworten (raw score) hervor. Demnach wird erwartet, dass in dieser Klasse 32% der Personen

	Klasse 1(0.59898)	Klasse 2(0.40102)
VAR1	0.64	0.67
VAR2	0.49	0.73
VAR3	0.37	0.64
VAR4	0.16	0.70
VAR5	0.24	0.42

Tabelle 4.2 Gegenüberstellung der Itemscores nach Variablen der beiden Klassen - KFT.dat

```
Final estimates in CLASS 1 of 2 with size  0.59898
=================================================

Expected Score Frequencies and Personparameters:

  score frequency  | person parameters and standard errors:

  Raw-  | Expected |   MLE-   |std. error|   WLE-   |std. error
  score | freq.    | estimate |    MLE   | estimate |    WLE
 -------|----------|----------|----------|----------|----------
      0 |    33.98 | ******** | ******** |   -2.976 |    1.760
      1 |    44.51 |   -1.716 |    1.208 |   -1.464 |    1.149
      2 |    41.22 |   -0.520 |    1.026 |   -0.465 |    1.023
      3 |    43.30 |    0.514 |    1.028 |    0.455 |    1.025
      4 |     0.66 |    1.717 |    1.212 |    1.465 |    1.153
      5 |    16.02 | ******** | ******** |    2.990 |    1.767

WLE estimates : Mean        =  -0.651 Var =   2.687 stdev =   1.639
                  marginal error variance =   1.689 stdev =   1.300
                       anova reliability  =   0.614
                     Andrichs reliability  =   0.372

WLE = Warm´s modified likelihood estimates,
MLE = Standard maximum likelihood estimates.

Raw-score     : Mean        =   1.890 Stdev =   1.432
```

Abb. 4.15 Final estimates class 1 - KFT.dat

keine einzige richtige Antwort (rawscore=0) geben können. 44% der Personen dieser Klassen können erwartungsgemäß zumindest eine richtige Antwort geben (rawscore=1). Auffällig ist, dass in dieser Klasse keine Personen erwartet werden, die insgesamt vier richtige Antworten geben können (rawscore=4), wohingegen erwartet wird, dass 18% der Befragten fünf richtige Antworten geben können (rawscore=5). Die Personenparameter für die Personen in dieser Klasse werden als MLE-Wert und als WLE-Wert dargestellt. Da bei den MLE-Schätzern keine Personenparameter für die extremsten Summenwerte (in diesem Fall für Personen mit rawscore=0 und rawscore=5) geschätzt werden können, werden diese estimates für eine Interpretation der Personenparametern nicht herangezogen. Für diesen Zweck könnten die WLE-Schätzern verwendet werden. ((Bühner, 2006, S. 363) So weist eine der ersten Klasse zugeordnete Person mit keiner richtigen Antwort eine Fähigkeit von -2,981 auf und ist somit als „nicht kognitiv fähig“ einzustufen. Hingegen weist eine Person mit fünf richtigen Antworten eine persönliche Fähigkeit von 2,996 auf und kann somit als „fähig“ im Sinne des kognitiven Fähigkeitstests eingestuft werden. Der „WLE-estimate mean“ gibt die durchschnittliche Fähigkeit der dieser Klasse zugeordneten Personen wieder. In diesem Fall liegt die durchschnittliche Fähigkeit der dieser Klasse zugeordneten Personen bei -0,592. Demnach sind die Personen dieser Klasse insgesamt als eher unterdurchschnittlich fähig einzustufen. Dies geht auch aus der durchschnittlichen Anzahl richtiger Antworten (raw-score mean) hervor, die in dieser Klasse rund 1,9 beträgt.

```
expected category frequencies and item scores:

   Item    |     Item‘s    | relative category
   label   | Score | Stdev |    frequencies
 ----------|-------|-------|    0    |    1
  VAR1     |  0.64 |  0.48 |  0.364  |  0.636
  VAR2     |  0.49 |  0.50 |  0.515  |  0.485
  VAR3     |  0.37 |  0.48 |  0.634  |  0.366
  VAR4     |  0.16 |  0.37 |  0.840  |  0.160
  VAR5     |  0.24 |  0.43 |  0.758  |  0.242

    Sum:   |  1.89

threshold parameters:  ordinal (partial credit) model

   item      |   item    |
   label     | location  | threshold parameters
-------------|-----------
  VAR1       | -1.54869
  VAR2       | -0.70653
  VAR3       | -0.04468
  VAR4       |  1.57870
  VAR5       |  0.72120
```

Abb. 4.16 category frequencies, item scores und threshold parameters class 1 - KFT.dat

Des weiteren wird die erwartete Häufigkeit der Antwortkategorien für die fünf Fragen dargestellt(Abb. 4.16). So wird erwartet, dass rund 65% der Personen dieser Klasse die erste Frage (VAR1) richtig beantworten können. Somit ist diese Frage innerhalb dieser Klasse als am einfachsten zu werten, was auch aus dem Itemparameter dieser Frage (-1,56) abgelesen werden kann. Die schwierigste Frage für diese Personen ist die vierte Frage (VAR4), die einen Itemparameter von 1,59 aufweist beziehungsweise nur von 17% der Personen in dieser Klasse richtig beantwortet werden kann.

```
item fit assessed by the Q-index

 itemlabel | Q-index |    Zq    | p(X>Zq)
-----------|---------|----------|-----------
 VAR1      |  0.0486 |  -0.4574 | 0.67631   | -....!Q...+ |
 VAR2      |  0.0606 |  -0.6766 | 0.75069   | -....!.Q..+ |
 VAR3      |  0.1576 |   0.7662 | 0.22178   | -.Q..!....+ |
 VAR4      |  0.1202 |   0.0426 | 0.48299   | -...Q!....+ |
 VAR5      |  0.1788 |   0.8966 | 0.18495   | -Q...!....+ |

                                  -?:p<0.05, +?:p>0.95
                                  -!:p<0.01, +!:p>0.99
```

Abb. 4.17 item fit class 1 - KFT.dat

Aus der „Item fit“ -Tabelle (Abb. 4.17) kann anhand des Q-index-Wertes abgelesen werden wie wahrscheinlich die Antwortmuster eines Items unter den gegebenen Modellparametern sind. Dieser Q-Wert kann grundsätzlich zwischen Null und Eins liegen, wobei Werte gleich Null besagen, dass bei diesem Item die Wahl einer Antwortkategorie exakt der Fähigkeitsausprägung einer Person entspricht. Liegt der Q-Index nahe bei 0,5 kann davon ausgegangen werden, dass bei diesem Item ein zufälliges Antwortmuster vorliegt und eventuell auszuschließen ist. Ein Q-Index nahe bei Eins deutet darauf

hin, dass sich das beobachtete Antwortmuster exakt umgekehrt zu dem vom Rasch-Modell erwarteten Antwortmuster verhält. Dies kann darauf hinweisen, dass das Item umzupolen ist. Als Normalwert für den Q-Index kann ein Bereich von 0,1 bis 0,3 gewertet werden. (Bühner, 2006, S.365f) und die dort angeführten Werke) Im vorliegenden Fall liegt der Q-Wert bei allen Items innerhalb des normalen Bereiches, weshalb von einem guten Item-Fit ausgegangen werden kann.

```
Final estimates in CLASS 2 of 2 with size  0.40102
==================================================

Expected Score Frequencies and Personparameters:

  score frequency  | person parameters and standard errors:

  Raw-  | Expected |   MLE-   |std. error|   WLE-   |std. error
  score | freq.    | estimate |    MLE   | estimate |    WLE
 _______|__________|__________|__________|__________|__________
      0 |    24.02 | ******** | ******** |   -2.650 |    1.672
      1 |     3.49 |   -1.545 |    1.152 |   -1.276 |    1.080
      2 |     4.78 |   -0.480 |    0.963 |   -0.425 |    0.959
      3 |     6.70 |    0.436 |    0.975 |    0.360 |    0.969
      4 |    59.34 |    1.551 |    1.183 |    1.273 |    1.110
      5 |    21.98 | ******** | ******** |    2.759 |    1.735

WLE estimates : Mean        =   0.569 Var =   3.330 stdev =   1.825
                marginal error variance =   1.839 stdev =   1.356
                     anova reliability  =   0.644
                   Andrichs reliability  =   0.448

WLE = Warm´s modified likelihood estimates,
MLE = Standard maximum likelihood estimates.

Raw-score     : Mean        =   3.162 Stdev =   1.775
```

Abb. 4.18 Final estimates class 2 - KFT.dat

Die zweite Klasse umfasst (siehe Abb. 4.18 rund 41% der Befragten. Von rund 26% dieser Personen wird erwartet, dass sie keine einzige der fünf Fragen richtig beantworten können (rawscore=0). Diese Personen weisen eine Fähigkeit von -2,64 auf. Hingegen wird erwartet, dass rund 60% der Befragten vier richtige Antworten (rawscore=4) geben können. Diese Personen weisen eine Fähigkeit von 1,27 auf. Jene 20% der Personen dieser Klasse, von denen erwartet wird, dass sie alle fünf Fragen richtig beantworten können (rawscore=5), weisen eine Fähigkeit von 2,75 auf. Die durchschnittliche Fähigkeit der Personen dieser Klasse, gemessen am WLE-estimate mean, betragt 0,475 und ist somit deutlich höher als in der ersten Klasse, wo der WLE-estimate mean -0,592 beträgt. Die durchschnittliche Anzahl der richtig beantworten Fragen (raw score mean) beträgt in dieser Klasse rund 3,1.

Die leichteste Frage für diese Personen ist Frage zwei (VAR2), die rund 71% der Personen dieser Klasse beantworten können (Abb. 4.19). Die Itemschwierigkeit dieser Frage beträgt -0,89. Die schwierigste Frage für die Personen dieser Klasse ist Frage fünf (VAR5), die leidiglich 40% der Befragten richtig beantworten können. Die Schwierigkeit dieses Items beträgt 1,33.

Aus der Q-Index-Tabelle (Abb. 4.20 geht hervor, dass die Items VAR2 und VAR4 deutlich über der zulässigen Obergrenze von 0,3 liegen, weshalb hier das Vorliegen zufälliger Antwortmuster nicht ausgeschlossen werden kann. Da sich die Antwortmuster dieser Items jedoch nicht signifikant von den zu erwarteten Antwortmustern unterscheiden, kann auf einen Ausschluss dieser Items verzichtet werden. (Bühner, 2006, S.366) In Bezug auf die zweite Klasse kann zusammenfassend festgehalten werden, dass diese Klasse eher überdurchschnittlich fähige Personen beinhaltet.

```
expected category frequencies and item scores:

   Item    |     Item's    | relative category
   label   | Score | Stdev |    frequencies
 ----------|-------|-------|    0    |    1
  VAR1     |  0.67 |  0.47 |  0.329  |  0.671
  VAR2     |  0.73 |  0.44 |  0.270  |  0.730
  VAR3     |  0.64 |  0.48 |  0.358  |  0.642
  VAR4     |  0.70 |  0.46 |  0.300  |  0.700
  VAR5     |  0.42 |  0.49 |  0.581  |  0.419

    Sum:   |  3.16

threshold parameters:  ordinal (partial credit) model

   item      |   item    |
   label     | location  | threshold parameters
-------------|-----------
  VAR1       | -0.12680
  VAR2       | -0.89137
  VAR3       |  0.13855
  VAR4       | -0.46740
  VAR5       |  1.34702
```

Abb. 4.19 category frequencies, item scores und threshold parameters class 2 - KFT.dat

```
item fit assessed by the Q-index

 itemlabel | Q-index |    Zq    | p(X>Zq)
-----------|---------|----------|-----------
 VAR1      |  -0.0000 |   0.0291 | 0.48839   | -...Q!....+ |
 VAR2      |  0.3607 |  -0.0587 | 0.52340   | -....Q....+ |
 VAR3      |  0.1424 |  -0.0192 | 0.50766   | -....Q....+ |
 VAR4      |  0.4371 |   0.0824 | 0.46715   | -...Q!....+ |
 VAR5      |  -0.0000 |  -0.0356 | 0.51420   | -....Q....+ |

                                  -?:p<0.05, +?:p>0.95
                                  -!:p<0.01, +!:p>0.99
```

Abb. 4.20 item fit class 1 - KFT.dat

Die „person fit index descriptives“ (Abb. 4.21) geben Auskunft über die Anpassungsgüte des Mixed Rasch Modells an die Antwortmuster der Befragten. Die dargestellten Werte stellen z-Werte dar, wobei Werte kleiner -1,96 oder größer als +1,96 als kritisch anzusehen sind, wohingegen Werte innerhalb dieser Intervalle auf eine gute Modellanpassung hinweisen. (Bühner, 2006, S. 367) Im Beispiel kann folglich von einer guten Modellanpassung ausgegangen werden. Aus der Tabelle „statistics of expected class membership“ können zunächst die erwarteten Klassengrößen abgelesen werden. So wird erwartet, dass rund 57% der Personen aufgrund ihrer Antwortmuster der ersten Klasse zugeordnet werden, wobei die Wahrscheinlichkeit einer richtigen Zuordnung rund 93% beträgt. In Bezug auf die zweite Klasse wird erwartet, dass dieser rund 28% Der Befragten zugewiesen werden, wobei die Wahrscheinlichkeit einer richtigen Zuordnung lediglich 76% beträgt. Die Treffsicherheit der Klassenzuordnung ist somit als eher schlecht einzuordnen. Die Beurteilung der Modellgüte anhand der im Output dargestellten Fit-Werte ist auch in diesem Fall nicht sinnvoll. Diese Beurteilung kann anhand eines parametrischen Bootstrappings durchgeführt werden. Um dies in WinMiRa durchzuführen ist unter <Job Definition> <Bootstrap GoF> auszuwählen. Im vorliegenden Beispiel wurden die

```
person fit index descriptives:

mean       :      -0.0521870
std.dev.   :       0.9837442

skewness   :      -0.7135799
kurtosis   :      -0.3993372


statistics of expected class membership:

      | exp.  | mean  |
class | size  | prob. |   1   |   2   |
------|-------|-------|-------|-------|
    1 | 0.543 | 0.780 | 0.780| 0.220|
    2 | 0.290 | 0.970 | 0.030| 0.970|
```

Abb. 4.21 person fit descriptives - KFT.dat

Programmvoreinstellungen übernommen. Anschließend ist das Modell durch <Start><Start Job> erneut zu berechnen. Der Output wurde nun um die Ergebnisse des Bootstrapping ergänzt (Abb. 4.22).

```
Parametric Bootstrap estimates for Goodness of Fit:

No.:      Satlik      LogLik          LR     CressieRead     Pearson X^2          FT

  1    -779.595    -792.705      26.221          30.864          39.111     32.3025

 40    -761.238    -769.219      15.963          20.484          26.258     17.8666

                Z:                1.011           0.665           0.285      0.5376
          P(X>Z):                 0.156           0.253           0.388      0.2954
            Mean:                16.442          16.409          18.060     21.4634
           Stdev:                 4.704           5.730           8.045      6.6698
 p-values (emp. PDF):             0.125           0.200           0.300      0.3250
```

Abb. 4.22 parametrischer Bootstrap - KFT.dat

Von Interesse sind dabei vorrangig die Cressie-Read-Prüfgröße (CressieRead) und der Pearson χ^2-Wert beziehungsweise deren p-Werte (p-values). Signifikante Prüfwerte signalisieren, dass das untersuche Modell zu verwerfen ist. Da dies in diesem Beispiel nicht der Fall ist, kann das Mixed Rasch Modell grundsätzlich angenommen werden, was eine gute Anpassung der erwarteten an die beobachteten Antwortmuster der beiden Klassen bescheinigt.

Der graphischen Output kann wie bei der Latent Class Analyse interpretiert werden und wird daher nicht gesondert angeführt.

Abschließend kann festgehalten werden, dass anhand des Mixed Rasch Modells, wie auch bereits im Latent Class Modell zwischen „fähigen“ und „nicht fähigen“ Personen im Sinne des kognitiven Fähigkeitstest unterschieden werden kann. Jedoch erweitert dieses Modell die Aussagen der Latent Class Analyse dahingehend, dass die Personen innerhalb der unterschiedlichen Klassen nicht mehr dieselbe Lösungswahrscheinlichkeit bei allen Fragen aufweisen müssen, sondern durchaus unterschiedliche Lösungswahrscheinlichkeiten der Fragen möglich sind. Dadurch ist es möglich, auch innerhalb der jeweiligen Klassen zwischen „fähigeren“ und „weniger fähigeren“ Personen zu unterscheiden.

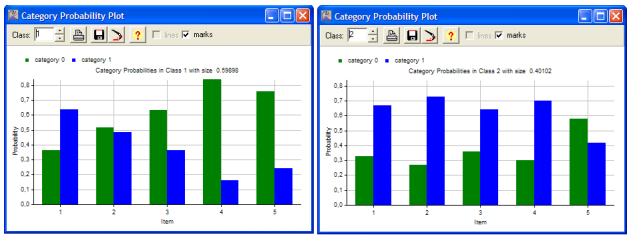

Category Probability Plot
Category Probabilities in Class 1 with size 0.59898
Category Probability Plot
Category Probabilities in Class 2 with size 0.40102

Abb. 4.23 Category Probability Plot - KFT.dat

Inwiefern das Mixed Rasch Modell mit zwei Klassen nun höhere Erklärungskraft hat als das vorher betrachtete Latent Class Modell mit zwei Klassen kann anhand eines Vergleichs der Informationskriterien (Tab. 4.3.2) festgestellt werden.

	AIC	BIC	CAIC
Latent Class Modell mit zwei Klassen	1.723,1	1.763,8	1.774,8
Mixed Rasch Modell mit zwei Klassen	1.719,0	1.779,20	1.796,0

Tabelle 4.3 Informationskriterien: AIC,BIC,CAIC - KFT.dat

So weist das Latent Class Modell mit zwei Klassen die geringeren BIC- und CAIC-Werte auf, weshalb diese Lösung zu favorisieren ist.

4.4 Conclusio

Das Programm ist intuitiv zu bedienen und verfügt über sehr gut auskommentierte Literatur (siehe Kapitel 4.2.3.2). Somit ist der Einarbeitungsaufwand in das Programm überschaubar. Die grafische Bedienungsoberfläche erleichtert das Aufrufen der Kommandos / Prozeduren. Die übersichtlichen Grafiken verdeutlichen auf sehr einfache Art und Weise die Personen- und Itemparameter der Modelle. Die Schnittstelle für den Import / Export von .sav und .csv Files funktioniert einwandfrei. Alles in allem kann WinMiRa aufgrund der beschriebenen Spezifika durchaus als geeignetes Programmpaket für den Einstieg in Rasch-Analysen bezeichnet und weiterempfohlen werden.

Kapitel 5
BILOG-MG und MULTILOG

Marco J. Maier und Konradin Maier

Zusammenfassung In diesem Kapitel werden die Softwarepakete BILOG-MG und MULTILOG vorgestellt, die zur Analyse dichotomer, bzw. polytomer Daten verwendet werden. Beide Programme sind für Microsoft Windows ausgelegt und verfügen sowohl über ein code-orientiertes als auch eine grafisches Benutzerinterface, was sie auch für IRT-Neulinge interessant macht.

5.1 Einleitung

BILOG-MG (Zimowski et al., 1996), eine Erweiterung von BILOG 3 (Mislevy und Bock, 1990) für mehrere Gruppen, ist für die Analyse binärer Items ausgelegt, wobei auch Antwortformate wie Multiple-Choice als richtig/falsch codiert verwendet werden können. Zur Auswertung stehen 1-, 2- oder 3-Parameter-Logistischen Modellen zur Verfügung. Zusätzlich können Phänomene wie *differential item functioning* (DIF), *item parameter drift* (DRIFT) oder *variant item analysis* (VARIANT) untersucht werden. Um die Vergleichbarkeit von Scorings bei neuen Testformaten oder anderen Personengruppen zu gewährleisten liegen auch hier entsprechende Verfahren vor.

MULTILOG (Thissen, 1991) bietet im Vergleich zu BILOG-MG dieselben Verfahren für binäre Items, darüber hinaus sind hier jedoch auch eine Vielzahl an Modellen für kategorielle Daten (ordinal und nominal), inklusive DIF-Routinen, implementiert.

5.2 Modelle

5.2.1 BILOG-MG

Abgesehen von herkömmlichen IRT-Modellen hat BILOG-MG zusätzlich Modelle zur Mehrgruppenanalyse. Die zentrale Annahme bei mehreren Gruppen ist, dass Personen aus unterschiedlichen Populationen stammen, die jeweils normalverteilt sind. Zur Schätzung werden somit die Itemparameter und die Parameter der gruppenspezifischen Verteilungen verwendet. Zusätzlich bietet BILOG-MG spezielle Analysemöglichkeiten für DIF, DRIFT und VARIANT Modelle.

5.2.1.1 1PLM

Das 1PLM (1-Parameter-Logistisches Modell) modelliert eine richtige Antwort der Person v auf Item i ($x_{vi} = 1$) folgendermaßen:

WU Wien, Institut für Statistik und Mathematik • marco.maier@wu.ac.at
WU Wien, Institut für Informationswirtschaft • konradin.maier@wu.ac.at

$$\Pr(x_{vi} = 1|\theta_v, \beta_i, \alpha) = \frac{\exp(\alpha(\theta_v - \beta_i))}{1 + \exp(\alpha(\theta_v - \beta_i))} \tag{5.1}$$

wobei θ_v der Fähigkeitsparameter der Person, β_i der Schwierigkeitsparameter des Items und α ein Skalierungsparameter (Diskrimination) sind. Im Gegenteil zum herkömmlichen Raschmodell wird hier ein Itemdiskriminationsparameter geschätzt, der jedoch für alle Items gleich ist.

5.2.1.2 2PLM und 3PLM

Zu diesem Modell gibt es noch zwei Erweiterungen, die in BILOG-MG umgesetzt sind, nämlich das 2PLM, das zusätzlich einen Itemdiskriminationsparameter α_i pro Item i enthält und das 3PLM, welches die Ratewahrscheinlichkeit pro Item durch einen entsprechenden Rateparameter γ_i modelliert.

Formal haben 2- und 3PL-Modelle die folge Form:

$$\Pr(x_{vi} = 1|\theta_v, \beta_i, \alpha_i) = \frac{\exp(\alpha_i(\theta_v - \beta_i))}{1 + \exp(\alpha_i(\theta_v - \beta_i))} \tag{5.2}$$

$$\Pr(x_{vi} = 1|\theta_v, \beta_i, \alpha_i, \gamma_i) = \gamma_i + (1 - \gamma_i)\frac{\exp(\alpha_i(\theta_v - \beta_i))}{1 + \exp(\alpha_i(\theta_v - \beta_i))} \tag{5.3}$$

In Abbildung 5.1 sind die sog. Itemcharakteristikkurven (*Item Characteristic Curve*; ICC) der drei Modelle abgebildet. Der Schwierigkeitsparameter β_i jedes Items wird hier durch eine vertikalen Linie angezeigt. Die Items liegen von links nach rechts bei $\beta_i = -4, -2, 0$ und $+3$. Die drei rechten Items wären typische Vertreter eines 1-, oder 2PL-Modells mit unterschiedlichen Diskriminationen. Die Aufgabe bei $\beta = 0$ hat einen Anstieg von genau $\alpha = 1$, was identisch zu den ICCs eines Raschmodells ist. Die schwierige Frage ganz rechts besitzt eine große Diskriminationsfähigkeit von $\alpha = 2$, d.h. der Anstieg ist steil während das Item bei $\beta = -2$ mit $\alpha = 0.5$ flach verläuft und somit schlecht diskriminiert.

Die Kurve links außen zeigt exemplarisch eine ICC eines 3PLM, bei dem $\beta = -4$, $\alpha = 1.5$ und $\gamma = .2$, d.h. es besteht eine Ratewahrscheinlichkeit von 20% (bspw. bei einem Multiple-Choice Test mit 5 Alternativen, von denen genau eine richtig ist). Dies führt dazu, dass die Kurve mit $\theta \to -\infty$ nicht asymptotisch gegen 0, sondern gegen γ – in diesem Fall 0.2 – geht.

5.2.2 MULTILOG

5.2.2.1 1-, 2- und 3-Parameter Logistische Modelle

MULTILOG beherrscht, wie auch BILOG-MG die 1-, 2- und 3-Parameter-Logistischen Modelle für deren Erklärung wir auf die Abschnitte 5.2.1.1 sowie 5.2.1.2 in diesem Kapitel verweisen wollen.

5.2.2.2 Graded Response Modell (Samejima)

Das *graded response model* von Samejima (1969) für ordinale Antworten mit k Antwortkategorien (von 1 bis m, wobei höhere Kategorien mit höheren Fähigkeitswerten θ einhergehen) ist in der logistischen Form wie folgt definiert:

$$\Pr(x_{vi} = k|\theta_v, \alpha, \beta_i k) = \qquad \Pr^\star(k) \qquad - \Pr^\star(k+1) \tag{5.4}$$

$$\Pr(x_{vi} = k|\theta_v, \alpha, \beta_i k) = \frac{1}{1+\exp(-D\alpha_i(\theta-\beta_i+\tau_k))} - \frac{1}{1 + \exp(-D\alpha(\theta - \beta_i + \tau_{k+1}))} \tag{5.5}$$

Folgene Einschränkungen gelten: $\Pr^\star(1) = 1$, $\Pr^\star(m+1) = 0$. Inhaltlich ist α ein globaler Diskriminationsparameter, β_i der Itemlokationsparameter und τ_k der Schwellenparameter für Kategorie k.

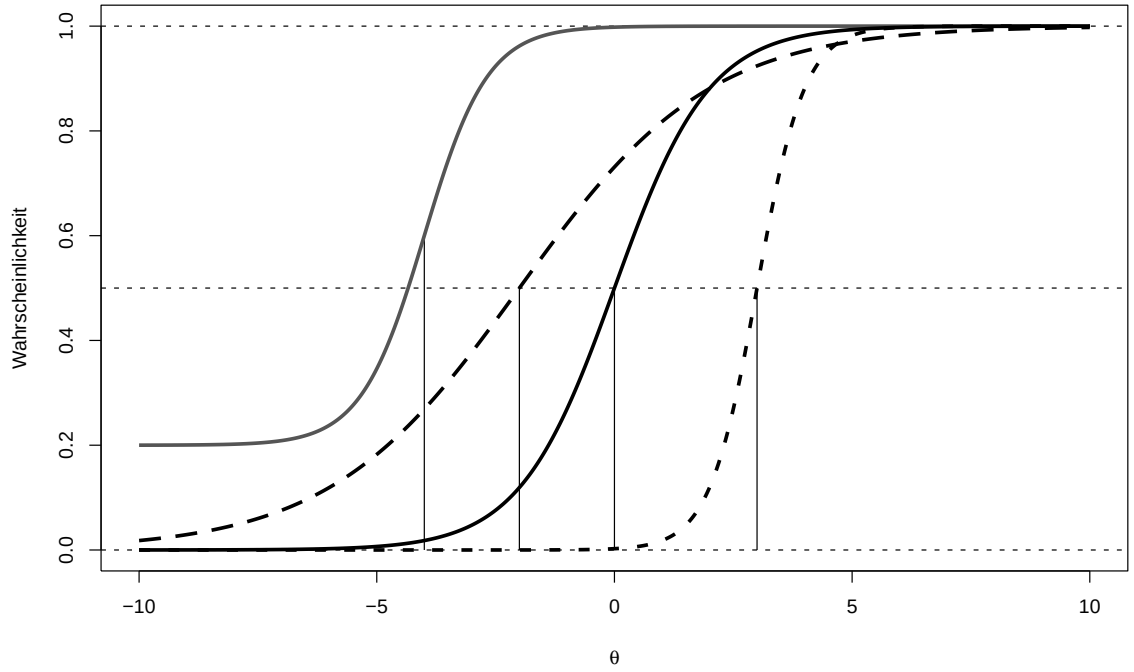


Abb. 5.1 Exemplarische ICCs der 1-, 2- und 3PL-Modelle

$\Pr^\star(k)$ ist die Wahrscheinlichkeit, dass eine Antwort in Kategorie k oder höher liegt, wodurch sich die Nebenbedingungen $\Pr^\star(1) = 1$ und $\Pr^\star(m+1) = 0$ erklären.

5.2.2.3 Nominal Response Modell (Bock)

Das *nominal response model* von Bock (1972) verrechnet Daten, deren Antwortalternativen von 1 bis m laufen. Wie der Name schon sagt müssen die Antwortkategorien der Items hier nicht geordnet sein, sondern können auch *qualitativ* (nominal) sein.

$$\Pr(x = k|\theta, \alpha, \beta) = \frac{\exp(\alpha_k\theta + \beta_k)}{\sum_{i=1}^{m} \exp(\alpha_i\theta + \beta_i)} \tag{5.6}$$

Jede Kateogrie hat einen Diskriminations (α) und Schwellenparameter (β) und die Lösungswahrscheinlichkeit einer Kategorie ergibt sich aus dem Quotienten der Kategorie und der Summe aller Kategorien.

5.2.2.4 Multiple Response und Multiple Choice Modelle

Diese Modelle wurden von Thissen und Steinberg (1984) vorgeschlagen und sind Modifikationen von Samejimas *graded response* Modell (siehe Abs. 5.2.2.2). Die Wahrscheinlichkeit, dass eine Antwort x in Kategorie k fällt berechnet sich für das *multiple response model* folgendermaßen:

$$\Pr(x = k) = \frac{h^\star \exp(\alpha_k\theta + \beta_k) + hd_k \exp(\alpha_1\theta + \beta_1)}{\sum_{i=1}^{m+1} \exp(\alpha_i\theta + \beta_i)} \tag{5.7}$$

Der Ausdruck $\alpha_k\theta+\beta_k$ ist eine andere Parametrisierung des 2PLM-Kerns $\alpha(\theta-\beta)$ und d_k ist der Anteil an Testpersonen, die zwar die Antwort nicht wissen, aber dennoch antworten. Durch unterschiedliche Werte von $h^\star$ und h kann man andere Modelle spezifizieren, wobei wir hier auf das multiple-choice Modell näher eingehen wollen.

Hat man mehr als zwei Kategorien ($m > 2$) und setzt man $h = 1$, so wird aus der obigen Formel das *multiple-choice Modell*:

$$\Pr(x = k) = \frac{\exp(\alpha_k\theta + \beta_k) + d_k \exp(\alpha_1\theta + \beta_1)}{\sum_{i=1}^{m+1} \exp(\alpha_i\theta + \beta_i)} \tag{5.8}$$

Hier muss man beachten, dass die Antwortkategorien bei 2 beginnen und demnach bis $m+1$ laufen, da Kategorie 1 für die latente „weiß nicht" Kategorie steht.

Zur Identifikation der Parameter müssen noch Einschränkungen gesetzt werden – entweder als Nullsetzung der Itemparameter $\alpha_1 = \beta_1 = 0$, oder als Summe-Null-Parametrisierung $\sum \alpha_k = \sum \beta_k = 0$. Alternativ dazu kann man auch mit Einschränkungen der d_k-Parameter arbeiten.

5.3 Parameterschätzung

Da sich die Parameterschätzmethoden der beiden behandelten Programme nur geringfügig unterscheiden wird hier auf eine abschnittsweise Trennung verzichtet und stattdessen an den entsprechenden Stellen auf Unterschiede verwiesen.

5.3.1 Itemparameter

Zur Schätzung der Itemparameter wird die sog. *marginal maximum likelihood* (MML) Methode (siehe Baker und Kim, 2004) verwendet. Im Gegensatz anderen Schätzverfahren, werden die Personenparameter hier mit einer bekannten Verteilungsfunktion modelliert (meist nimmt man $\theta \sim \mathcal{N}(0,1)$ an), was jedoch in der Schätzgleichung zu einem Integral führt, das über eine Gauß-Hermite-Quadratur approximiert werden muss. Die Anzahl der Quadraturpunkte ist frei wählbar, wobei die Autoren als Maximum $2\sqrt{\text{Itemanzahl}}$ vorschlagen. In weiterer Folge werden der EM-Algorithmus und Newton-Gauß-Verfahren zur Optimierung eingesetzt (für eine detailliertere Darstellung siehe Bock und Aitkin, 1981).

5.3.1.1 Marginal Maximum A Posteriori

Die 2- und 3PL-Modelle haben starke Ähnlichkeiten zur unidimensionalen Faktorenanalyse, wodurch sog. *Heywood cases* auftreten können, die zur Folge haben, dass die Diskrimination einzelner Parameter gegen Unendlich gehen. Um diesem Phänomen entgegenzuwirken kommt eine Bayesianische Strategie namens *marginal maximum a posteriori* (MMAP) zum Einsatz, bei der als *a priori Verteilung* der Parameter eine lognormale Verteilung angenommen wird (da Diskriminationen positiv sein müssen). Beim 3PLM kommt dieser Ansatz auch bei den Rateparametern zum Einsatz, wo bei sich hier für das offene Intervall $(0,1)$ die Betaverteilung anbietet.

5.3.2 Personenparameter

5.3.2.1 Maximum Likelihood

Zur Schätzung der Itemparameter mittels *maximum likelihood* (ML) werden die Fähigkeitsparameter über einen Fisher-scoring Algortihmus so angepasst, dass die Likelihood maximiert wird. Im Gegensatz zu den beiden Bayesianischen Verfahren ist es nicht möglich Parameter für Antwortmuster, bei denen alle Items gelöst bzw. nicht gelöst wurden, zu schätzen.

5.3.2.2 Expected A Posteriori

Ein Bayesianisches Schätzverfahren ist mit *expected a posteriori* (EAP) implementiert. Hier werden die Fähigkeitsparameter als Mittelwerte der *a-posteriori-Verteilung* von θ unter gegebenen Antwortpatterns geschätzt. Die Standardfehler sind hier am geringsten, wobei Schätzer einen Bias zum Populationsmittel aufweisen.

5.3.2.3 Maximum A Posteriori

Die *maximum a posteriori* (MAP) Schätzer sind ähnlich den EAP, nur dass hier, statt dem Erwartungswert, der Modus der jeweiligen a posteriori Verteilung berechnet wird.

5.4 Bedienung der Software und Anwendungsbeispiele

Von der Bedienung her sind sich beide Programme sehr ähnlich, daher werden wir den Fokus auf BILOG-MG richten und danach Abweichungen und Erweiterungen von MULTILOG ergänzen. Letztlich wird eine Datei mit Spezifikationen benötigt, um Analysen durchzuführen. Komplexere Verfahren oder fortgeschrittene Anwendungen wie Parameterconstraints sind ausschließlich durch manuelle Angaben möglich, jedoch ist ein – vor allem für unerfahrene Personen – einigermaßen intuitives Menü vorhanden, das beim Erstellen einer korrekten Steuerdatei hilft.

5.4.1 BILOG-MG

5.4.1.1 Syntax und Vorbereitung

Startet man BILOG-MG, so öffnet sich eine schlichte grafische Benutzeroberfläche mit einem Texteditor für die `.blm`-Dateien, die den späteren Programmablauf steuern (siehe Abb. 5.2). Mit `File` und `New` wählt man einen Speicherort für seine Syntaxdatei. Es sind nun einige Dialogfenster auszufüllen, damit BILOG-MG eine entsprechende Analysesyntax generieren kann.

In der Menüleiste kann man mittels `Setup` und `General...` den allgemeinen Dialog mit Spezifikationen aufrufen. Es erscheint ein Dialog mit vier Reitern (`Job Description`, `Model`, `Response` und `Labels`) in denen man zuerst die Anzahl von Personen, Gruppen, Items und Subtests angibt. Danach spezifiziert man, wie in Abb. 5.3 ersichtlich, das auszuführende Modell (BILOG-MG bietet abgesehen von der verbreiteten Logit-Parametrisierung auch die Möglichkeit eine normale *Response Function Metric* zu wählen).

Es folgt die Definition der Responses, wobei man hier die Anzahl der Antwortalternativen und die richtigen Antwortmuster angibt, und die Behandlung fehlender Werte einstellt (es besteht die Möglichkeit einen fehlenden Wert als *teilrichtig* zu verrechnen indem er als $1/m$ repräsentiert wird,

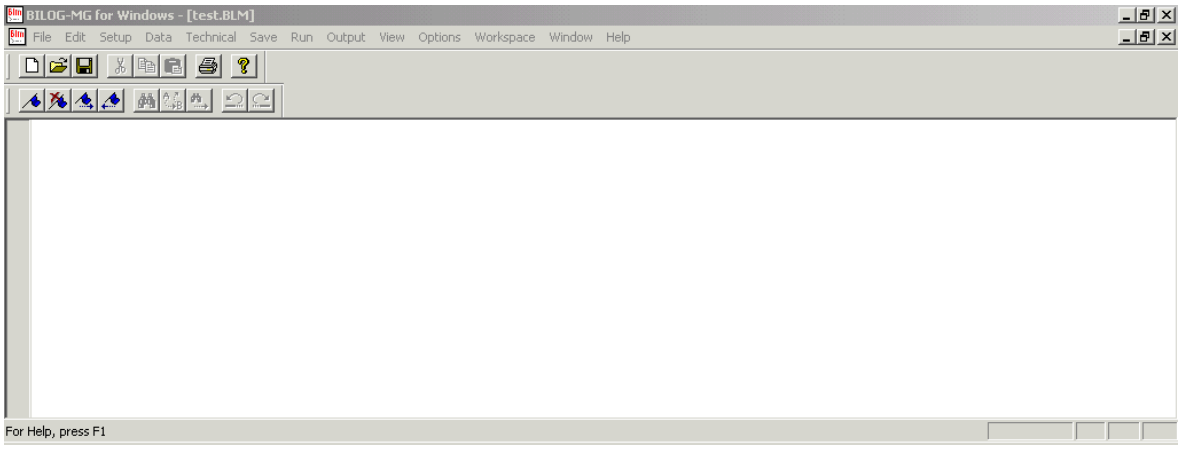

BILOG-MG for Windows - [test.BLM]
File Edit Setup Data Technical Save Run Output View Options Workspace Window Help
For Help, press F1

Abb. 5.2 BILOG-MG Oberfläche

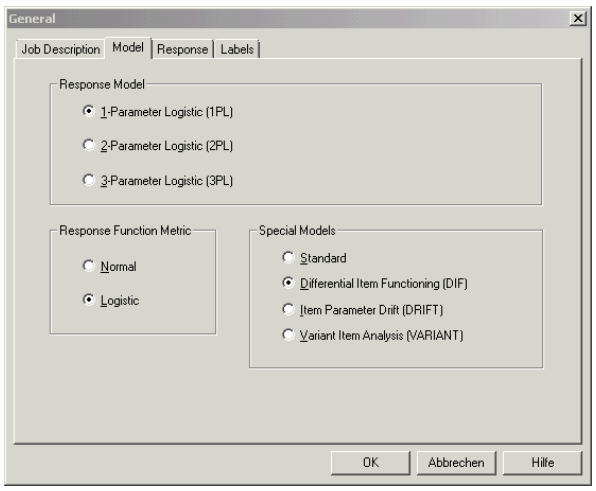


Abb. 5.3 Einstellungen und Auswahl des Analysemodells in BILOG-MG

wobei m die Anzahl der Alternativen ist). Zuletzt kann man noch passende Labels vergeben, die nach der Analyse im Output aufscheinen.

In einem zweiten Schritt wählt man unter `Setup` die Option `Item Analysis`, wobei sich nun ein Dialog mit drei Reitern (`Subtests`, `Subtest Items` und `Advanced`) öffnet. Wie der Name schon sagt kann man in den ersten zwei Abschnitten die Einstellungen für die Verrechnung der Subtests treffen, wobei diese nur erscheinen, wenn man im vorherigen Dialog die passende Anzahl an Subtests etc. ausgewählt hat. Der Reiter `Advanced` bietet viele technische Einstellungen, wie das Konvergenzkriterium, die maximale Anzahl an Iterationen für die EM und Newton Algorithmen oder die Anzahl der Punkte für die Gaußquadratur (siehe Abb. 5.4).

Unter `Setup` und `Test Scoring` kann man noch Einstellungen zur Schätzung der Personenparameter (z.B. ML, EAP oder MAP) treffen und ggf. eine Reskalierung der Werte vornehmen.

`Examinee Data` unter dem Menüeintrag `Data` öffnet einen weiteren Dialog in dem die Daten eingelesen werden (hierfür wird FORTRAN-Syntax verwendet, welche dem User durch Wizard jedoch abgenommen wird). Weiters kann man Fälle gewichten, oder auch nur ein Sample aller Beobachtungen für die Analyse verwenden. In `Item Keys` gibt man für jedes Item an, welche Antwort korrekt ist (`Answer Key`) und welche Codes für „Item nicht vorgegeben" bzw. „Fehlwert" stehen (`Not Presented Key` und `Omit Key`). Will man noch tiefgreifendere technische Einstellungen treffen kann man unter `Technical` beispielsweise Parameterstartwerte, Priorverteilungen und viele weitere Optionen setzen.

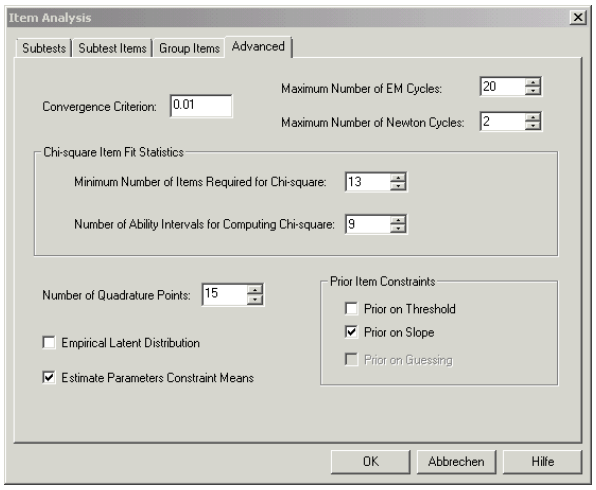


Abb. 5.4 Technische Einstellungen in BILOG-MG

Hat man alles eingestellt und angepasst wählt man unter `Run` den Punkt `Build Syntax`, wodurch BILOG-MG die getroffenen Optionen in Befehle übersetzt und diese in die eingangs angelegte Datei schreibt. In unserem Fall sieht diese folgendermaßen aus (kursive Elemente sind Kommentare zur Syntax):

```
Beispielhafte Analysesyntax - Titel          die Angabe eines Titels ist möglich
        MULTILOG-MG interpretiert die ersten beiden Zeilen nicht
>COMMENT         zusätzlich kann man noch einen längeren Kommentar beifügen
Daten aus Maier (2009);

>GLOBAL DFName = 'data.dat',          Pfad und Name des verwendeten Datensatzes
        NPArm = 1,          1-, 2- oder 3PLM
        LOGistic;          normale oder logistische Antwortfunktion
>LENGTH NITems = (12);          Anzahl der verwendeten Items
>INPUT NTOtal = 13,          Anzahl der eingelesenen Items
       NALt = 2,          Anzahl der Antwortalternativen
       NIDchar = 3,          Länge der Zeichenkette zur Personenidentifikation
       NGRoup = 2,          Anzahl der Gruppen
       DIF;          Analyse von Differential Item Functioning
>ITEMS ;
>TEST1 TNAme = 'logiktst',          Name des Tests (nützlich bei Subtests)
       INUmber = (1(1)5, 7(1)13);          verwendete Items, 1--5 und 7--13
>GROUP1 GNAme = 'maenner',          Name für die Untergruppe
        LENgth = 12,          Anzahl der Items
        INUmbers = (1(1)5, 7(1)13);          verwendete Items
>GROUP2 GNAme = 'frauen',          siehe oben
        LENgth = 12,
        INUmbers = (1(1)5, 7(1)13);
(3A1, 15X, I1, T5, 13A1)          Spezifikation der Datei in FORTRAN-Syntax
>CALIB NQPt = 30,          30 Punkte für die Gaußquadratur
       ACCel = 1.0000,
       CHIsquare = (13, 9);
>SCORE METhod = 1;
```

Mit `Run` und `Stats, Calibration and Scoring` gibt man die Anweisung gemäß der Syntax alle drei Phasen zu durchlaufen, worauf sich drei DOS-Fenster öffnen und die eigentlichen Berechnungen ausführen. Ist die Analyse erfolgreich verlaufen öffnet sich ein kleiner Dialog als Bestätigung. Nun kann man die Outputdateien aller drei Phasen ansehen (die jeweiligen Dateien haben die Endungen `.ph1`, `.ph2` und `.ph3`) und Plots anfordern.

5.4.1.2 Outputdateien

Im Output der ersten Phase findet man allgemeine Informationen zu den Daten und Einstellungen (Anzahl der Items, Labels, Gruppenspezifikationen etc.) sowie Itemstatistiken, die folgendermaßen aussehen:

```
ITEM STATISTICS FOR MULTIPLE GROUPS      LOGIKTST

                                                          ITEM*TEST CORRELATION
ITEM   NAME        #TRIED    #RIGHT    PCT     LOGIT    PEARSON  BISERIAL
-------------------------------------------------------------------------
   1   ITEM0001    204.0      88.0   0.431     0.28     0.325    0.409
   2   ITEM0002    204.0      85.0   0.417     0.34     0.273    0.345
   3   ITEM0003    204.0      57.0   0.279     0.95     0.120    0.161
   4   ITEM0004    204.0      68.0   0.333     0.69     0.353    0.457
   5   ITEM0005    204.0     104.0   0.510    -0.04     0.364    0.457
   6   ITEM0007    204.0      35.0   0.172     1.57     0.103    0.153
   7   ITEM0008    204.0      92.0   0.451     0.20     0.381    0.478
   8   ITEM0009    204.0      69.0   0.338     0.67     0.187    0.241
   9   ITEM0010    204.0      95.0   0.466     0.14     0.334    0.419
  10   ITEM0011    204.0      58.0   0.284     0.92     0.077    0.102
  11   ITEM0012    204.0      42.0   0.206     1.35     0.041    0.058
  12   ITEM0013    204.0      81.0   0.397     0.42     0.187    0.238
-------------------------------------------------------------------------
```

Für jede der definierten Gruppen wird eine solche Zusammenfassung erstellt, diese enthält den gesamten Datensatz. Man kann erkennen, dass jedes Item von 204 Personen bearbeitet wurde. Die Spalten `#RIGHT` und `PCT` enthalten die Anzahl bzw. Prozent richtiger Antworten. Logit sind die negativen logarithmierten Wettquotienten der Lösungswahrscheinlichkeit ($-\log[pct/(1-pct)]$) und stellt somit eine Art Schwierigkeitsschätzer dar. Zuletzt werden noch Pearson und punkt-biseriale Korrelationen zwischen Testscore und Itemantwort angegeben.

Die zweite Phase enthält die Kalibrierung und detaillierte Informationen über die Optimierung. Es folgen die Parameterschätzer aller Items und Gruppen, in unserem besteht die Ausgabe aus zwei vollen Sets an Itemparametern, da eine Analyse hinsichtlich Differential Item Function durchgeführt wurde.

```
MODEL FOR GROUP DIFFERENTIAL ITEM FUNCTIONING


GROUP   1    MAENNER ;  ITEM PARAMETERS AFTER CYCLE   8

ITEM      INTERCEPT    SLOPE    THRESHOLD   LOADING   ASYMPTOTE
-----------------------------------------------------------------
ITEM0001 |  -0.128  |   0.681  |   0.188  |   0.563  |   0.000
         |   0.283* |   0.078* |   0.415* |   0.065* |   0.000*
         |          |          |          |          |
ITEM0002 |   0.534  |   0.681  |  -0.783  |   0.563  |   0.000
         |   0.360* |   0.078* |   0.529* |   0.065* |   0.000*
         |          |          |          |          |
ITEM0003 |  -0.675  |   0.681  |   0.990  |   0.563  |   0.000
         |   0.405* |   0.078* |   0.595* |   0.065* |   0.000*
         |          |          |          |          |
ITEM0004 |   0.003  |   0.681  |  -0.004  |   0.563  |   0.000
         |   0.383* |   0.078* |   0.562* |   0.065* |   0.000*
         |          |          |          |          |
ITEM0005 |   0.672  |   0.681  |  -0.986  |   0.563  |   0.000
         |   0.320* |   0.078* |   0.469* |   0.065* |   0.000*
         |          |          |          |          |
ITEM0007 |  -1.302  |   0.681  |   1.912  |   0.563  |   0.000
         |   0.464* |   0.078* |   0.681* |   0.065* |   0.000*
         |          |          |          |          |
```

```
ITEM0008 |  -0.529  |   0.681  |   0.776  |   0.563  |   0.000
         |   0.360* |   0.078* |   0.529* |   0.065* |   0.000*
         |          |          |          |          |
ITEM0009 |  -0.671  |   0.681  |   0.984  |   0.563  |   0.000
         |   0.471* |   0.078* |   0.691* |   0.065* |   0.000*
         |          |          |          |          |
ITEM0010 |  -0.265  |   0.681  |   0.389  |   0.563  |   0.000
         |   0.324* |   0.078* |   0.476* |   0.065* |   0.000*
         |          |          |          |          |
ITEM0011 |  -1.298  |   0.681  |   1.905  |   0.563  |   0.000
         |   0.367* |   0.078* |   0.538* |   0.065* |   0.000*
         |          |          |          |          |
ITEM0012 |  -1.922  |   0.681  |   2.821  |   0.563  |   0.000
         |   0.354* |   0.078* |   0.519* |   0.065* |   0.000*
         |          |          |          |          |
ITEM0013 |  -0.813  |   0.681  |   1.193  |   0.563  |   0.000
         |   0.460* |   0.078* |   0.675* |   0.065* |   0.000*
-------------------------------------------------------------------
                                          * STANDARD ERROR
```

Die Spalten THRESHOLD und SLOPE sind Itemschwierigkeits- (β) und Itemdiskriminationsparameter (α). INTERCEPT und LOADING hingegen stellen die entsprechenden Parameter im Kontext der Faktorenanalyse dar, wobei $\texttt{INTERCEPT} = -\texttt{SLOPE} \cdot \texttt{THRESHOLD}$ und $\texttt{LOADING} = \texttt{SLOPE}/\sqrt{1 + \texttt{SLOPE}^2}$.

Der Parameter ASYMPTOTE wird erst mit dem 3PLM geschätzt und zeigt Rateparameter (γ) an. Pro Variable werden zwei Zeilen ausgegeben, in der oberen befinden sich die Parameterschätzungen, während darunter die jeweiligen Standardfehler (durch einen Asterisk markiert) angegeben sind. Für jede Subgruppe erhält man eine ähnliche tabellarische Ausgabe der Parameter.

Darunter finden sich die Mittelwerte aller Itemschwellenparameter der Gruppen, in unserem Fall jenen der Männer und Frauen. Die DIF-Analyse wird unter der Voraussetzung, dass die mittleren Schwellen in allen Gruppen gleich sind durchgeführt, daher werden jene der Frauen um die Differenz von 0.381 angepasst.

```
PARAMETER       MEAN  STN DEV
-----------------------------------
GROUP:   1  NUMBER OF ITEMS:    11
THRESHOLD      0.764    1.182
GROUP:   2  NUMBER OF ITEMS:    11
THRESHOLD      1.145    0.872
-----------------------------------



THRESHOLD MEANS

GROUP       ADJUSTMENT
------------------------
    1         0.000
    2         0.381
------------------------
```

Hier gibt BILOG die adjustierten Thresholdparameter aller Gruppen mit Standardfehlern aus.

```
MODEL FOR GROUP DIFFERENTIAL ITEM FUNCTIONING:
ADJUSTED THRESHOLD VALUES

  ITEM             GROUP          |   ITEM             GROUP
                 1          2     |                  1          2
---------------------------------+---------------------------------
   ITEM0001 |   0.188  |   0.213  |   ITEM0008 |   0.776  |  -0.061
            |   0.415* |   0.268* |            |   0.529* |   0.251*
            |          |          |            |          |
   ITEM0002 |  -0.783  |   0.534  |   ITEM0009 |   0.984  |   0.831
            |   0.529* |   0.260* |            |   0.691* |   0.256*
```

```
           |          |          |           |          |
   ITEM0003 |   0.990  |   1.384  |  ITEM0010 |   0.389  |  -0.102
           |   0.595* |   0.278* |           |   0.476* |   0.242*
           |          |          |           |          |
   ITEM0004 |  -0.004  |   1.097  |  ITEM0011 |   1.905  |   1.144
           |   0.562* |   0.269* |           |   0.538* |   0.266*
           |          |          |           |          |
   ITEM0005 |  -0.986  |  -0.178  |  ITEM0012 |   2.821  |   1.804
           |   0.469* |   0.241* |           |   0.519* |   0.309*
           |          |          |           |          |
   ITEM0007 |   1.912  |   2.429  |  ITEM0013 |   1.193  |   0.292
           |   0.681* |   0.327* |           |   0.675* |   0.264*
-------------------------------------------------------------------
 *STANDARD ERROR
```

Letztlich sehen wir eine Auflistung der Gruppendifferenzen für jedes Item mit entsprechenden Standardfehlern.

```
 MODEL FOR GROUP DIFFERENTIAL ITEM FUNCTIONING:
 GROUP THRESHOLD DIFFERENCES

   ITEM        GROUP    |   ITEM        GROUP
                2 - 1   |                2 - 1
-----------------------+-----------------------
   ITEM0001 |   0.025  |   ITEM0008 |  -0.838
           |   0.494* |            |   0.585*
           |          |            |
   ITEM0002 |   1.317  |   ITEM0009 |  -0.153
           |   0.589* |            |   0.737*
           |          |            |
   ITEM0003 |   0.394  |   ITEM0010 |  -0.491
           |   0.657* |            |   0.534*
           |          |            |
   ITEM0004 |   1.101  |   ITEM0011 |  -0.761
           |   0.623* |            |   0.600*
           |          |            |
   ITEM0005 |   0.808  |   ITEM0012 |  -1.017
           |   0.528* |            |   0.604*
           |          |            |
   ITEM0007 |   0.517  |   ITEM0013 |  -0.901
           |   0.756* |            |   0.725*
-----------------------------------------------
 *STANDARD ERROR
```

Durch die Schätzung eines DIF-Modells erhalten wir hier keinen Output zum Scoring, welches die dritte und letzte Phase darstellt.

5.4.1.3 Grafiken

BILOG besitzt eine Oberfläche für grafische Abbildungen der jeweiligen Analyse, die über ein allgemeines Menü produziert werden können (siehe Abb. 5.5). Man kann Item Characteristic Curves für ein oder mehrere Items (ICC, Matrix Plot), Informationskurven für einzelne oder alle Items (Information, Total Info), Kombinationen von ICC und Informationskurven für Items (ICC and Info) sowie Abbildungen der Fähigkeitsparameter (Bivariate Plot, Histogram) erzeugen.

In Abbildung 5.6 sehen wir die ICC (der β-Parameter ist durch ‚b' und einem Pfeil auf der x-Achse lokalisiert) und Informationskurve der ersten Aufgabe im Test.

Die Exportfunktion der Plotroutinen erlaubt die Speicherung von Grafiken als Metafiles (Windows und Enhanced Meta Files). In Abbildung 5.7 ist Item 3 der Subgruppe ‚Frauen' abgebildet, wobei wir hier 95% Intervalle für die beobachteten korrekten Antworten in den entsprechenden EAP gruppen eingezeichnet haben.

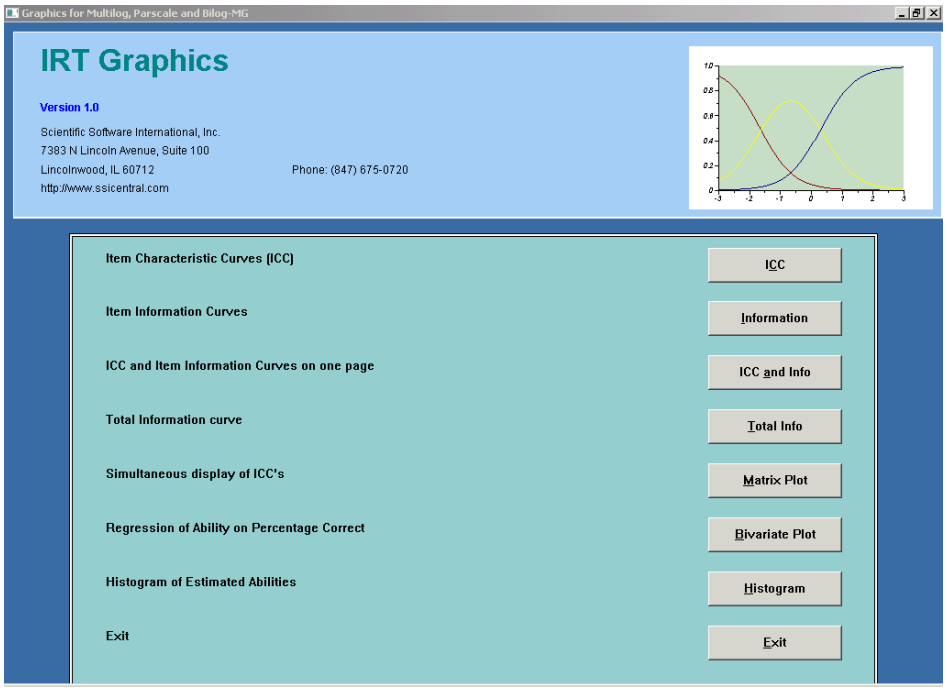


Abb. 5.5 Grafikmenü von BILOG

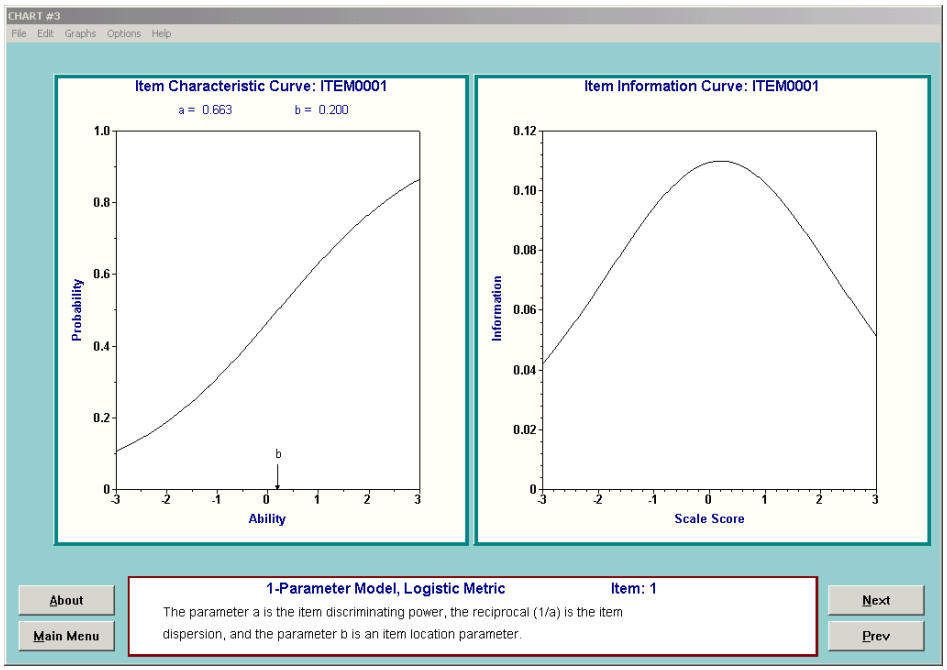


Abb. 5.6 Abbildung eines Items mittels Item Characteristic Curve und Item Information Curve

5.4.2 MULTILOG

Auch in MULTILOG sind diverse Unterstützungen zur Skriptgeneration implementiert, die unerfahrenen Personen den Umgang mit den diversen Befehlen abnehmen. In Abbildung 5.8 sieht man links den Einstiegsdialog, bei dem man die Steuerdatei anlegt und aus den Optionen „MML Item Parameter Estimation", „Fixed-theta Item Parameter Estimation", „MLE or MAP Computation" und „Blank MULTILOG Command File" wählen kann. Der rechte Screenshot zeigt den Auswahldialog für die verschiedenen Modelle inklusive einiger Einstellungen.

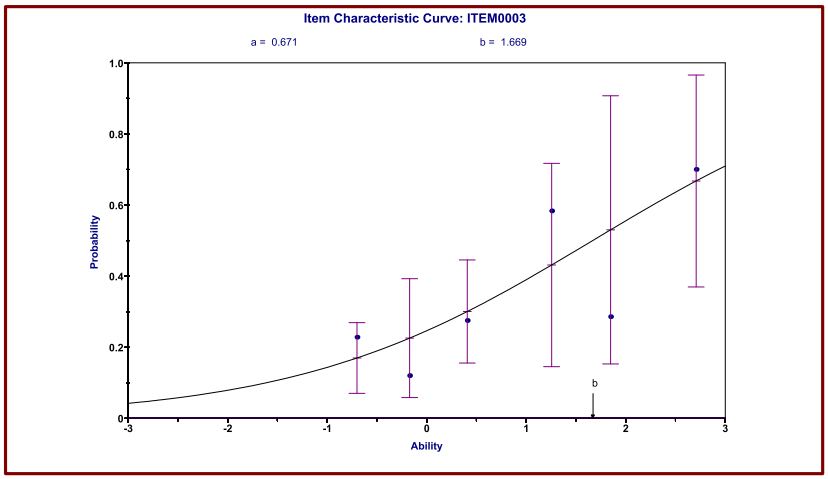


Abb. 5.7 ICC des dritten Items für die Subgruppe Frauen

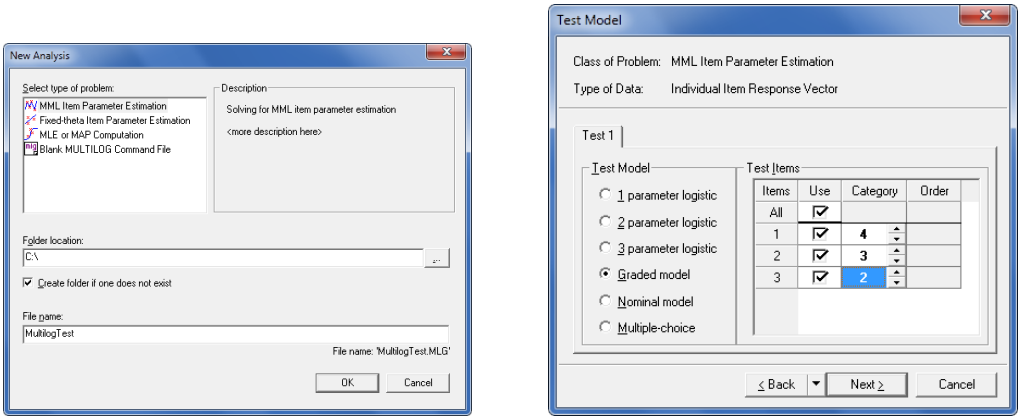


Abb. 5.8 Hilfsdialoge in MULTILOG

Zur Illustration ziehen wir ein Beispiel aus dem umfangreichen Angebot an Beispieldatensätzen heran, die in MULTILOG enthalten sind. Es handelt sich hierbei um eine Untersuchung von Klassen und O'Connor (1989), die zukünftige Gewaltdelikte anhand der Prädiktoren „Anzahl früherer (stationärer) Aufnahmen" und „Alter bei der ersten Aufnahme" vorhersagen versucht. Beide Variablen wurden in je vier ordinale Kategorien eingeteilt und zur Analyse herangezogen. Die zugrundeliegende Annahme ist, dass beide Indikatoren auf den Schweregrad psychischer Probleme schließen lassen.

Zur Auswertung wurde Samejimas *graded response model* verwendet und mit folgendem Code in MULTILOG berechnet:

```
Beispiel EXAMPL06.MLG - in MULTILOG enthalten          wieder werden die ersten 2 Zeilen ignoriert
Daten aus Klassen & O'Connor (1989)          hier kann man einen Titel einfügen
>PROBLEM   RANDOM,          wählt die MML Parameterschätzung
           PATTERNS,          Daten nicht als Einzelantworten, sondern in Antwortpatterns einlesen
           NITEMS=2, NGROUPS=1, NPATTERNS=16,          2 Items, 1 Gruppe, 16 Pattern
           DATA='EXAMPL06.DAT';          name der Datei mit dem Pattern
>TEST  ALL,          alle Items werden zur Analyse verwendet
       GRADED,          Samejimas graded response model
       NC=(4,4);          beide Items haben 4 Kategorien
>END;          Ende der Kommandodate
4          es gibt insgesamt vier Antwortcodes in den Daten, nämlich:
0123
11          kategorien für code = 0
22          kategorien für code = 1
33          kategorien für code = 2
44          kategorien für code = 3
(1X,2A1,F5.0)          die Datei „EXAMPL06.DAT" in FORTRAN-Format
```

Die dazugehörige Datei `EXAMPL06.DAT` sieht gemäß der Spezifikation `(1X,2A1,F5.0)` (Spalte 1 weglassen, Spalten 2–3 als einstellige alphanumerische Spalten interpretieren, Spalten 4–8 als fünfstellige Zahl einlesen) folgendermaßen aus:

```
00   28
01   15
02    8
03    5
10   35
11   23
12   12
13   15
20   43
21   35
22   19
23   29
30    6
31   14
32    6
33   11
```

Nachdem das Modell geschätzt wurde liefert MULTILOG nun, im Gegensatz zu BILOG, lediglich eine große Outputdatei die hier auszugsweise präsentiert wird. Wie in BILOG werden zu Beginn die Eingaben der Kommandodatei detailliert wiederholt und auch die technischen Einstellungen werden ausgegeben, wie man hier sieht:

```
DATA PARAMETERS:
 NUMBER OF LINES IN THE DATA FILE:   16
 NUMBER OF CATEGORICAL-RESPONSE ITEMS:   2
 NUMBER OF CONTINUOUS-RESPONSE ITEMS, AND/OR GROUPS:   1
 TOTAL NUMBER OF "ITEMS" (INCLUDING GROUPS):   3
 NUMBER OF CHARACTERS IN ID FIELDS:  0
 MAXIMUM NUMBER OF RESPONSE-CODES FOR ANY ITEM:  4
 THE MISSING VALUE CODE FOR CONTINUOUS DATA:  9.0000
 RESPONSE-PATTERN FREQUENCIES WILL BE READ
 THE DATA WILL BE STORED IN MEMORY

ESTIMATION PARAMETERS:
 THE ITEMS WILL BE CALIBRATED--
   BY MARGINAL MAXIMUM LIKELIHOOD ESTIMATION
 MAXIMUM NUMBER OF EM CYCLES PERMITTED:  25
 NUMBER OF PARAMETER-SEGMENTS USED IS:   2
 NUMBER OF FREE PARAMETERS IS:    8
 MAXIMUM NUMBER OF M-STEP ITERATIONS IS   4 TIMES
   THE NUMBER OF PARAMETERS IN THE SEGMENT
 THE M-STEP CONVERGENCE CRITERION IS: 0.000100
 THE EM-CYCLE CONVERGENCE CRITERION IS: 0.001000
 THE RK CONTROL PARAMETER (FOR THE M-STEPS) IS:  0.9000
 THE RM CONTROL PARAMETER (FOR THE M-STEPS) IS:  1.0000
 THE MAXIMUM ACCELERATION PERMITTED IS:  0.0000
 THETA-GROUP LOCATIONS WILL REMAIN UNCHANGED
```

Es folgen die Itemparametrschätzer des Modells. `A` steht für den Diskriminationsparameter α, `B(`k`)` für den Schwellenparameter von Kategorie k oder höher.

```
ITEM   1:       4 GRADED CATEGORIES
       P(#) ESTIMATE (S.E.)
A         1    0.87  (0.16)
B( 1)     2   -1.95  (0.37)
B( 2)     3   -0.19  (0.17)
B( 3)     4    2.57  (0.48)

@THETA:      INFORMATION:   (Theta values increase in steps of 0.2)
-3.0 - -1.6  0.159  0.172  0.184  0.194  0.203  0.211  0.217  0.221
```

```
-1.4 -  0.0  0.225  0.227  0.228  0.228  0.228  0.227  0.225  0.223
 0.2 -  1.6  0.220  0.217  0.215  0.212  0.210  0.209  0.209  0.209
 1.8 -  3.0  0.209  0.209  0.208  0.206  0.203  0.197  0.190

 OBSERVED AND EXPECTED COUNTS/PROPORTIONS IN
 CATEGORY(K):  1      2      3      4
 OBS. FREQ.     56     85    126     37
 OBS. PROP.  0.1842 0.2796 0.4145 0.1217
 EXP. PROP.  0.1844 0.2793 0.4144 0.1219


ITEM   2:       4 GRADED CATEGORIES
       P(#) ESTIMATE (S.E.)
A         5    1.01  (0.16)
B( 1)     6   -0.64  (0.17)
B( 2)     7    0.77  (0.16)
B( 3)     8    1.66  (0.24)

@THETA:      INFORMATION:   (Theta values increase in steps of 0.2)
-3.0 - -1.6  0.079  0.093  0.109  0.127  0.146  0.166  0.187  0.208
-1.4 -  0.0  0.228  0.247  0.263  0.277  0.289  0.298  0.304  0.309
 0.2 -  1.6  0.313  0.315  0.316  0.315  0.313  0.309  0.303  0.294
 1.8 -  3.0  0.282  0.267  0.250  0.231  0.210  0.189  0.168

 OBSERVED AND EXPECTED COUNTS/PROPORTIONS IN
 CATEGORY(K):  1      2      3      4
 OBS. FREQ.    112     87     45     60
 OBS. PROP.  0.3684 0.2862 0.1480 0.1974
 EXP. PROP.  0.3691 0.2857 0.1475 0.1976
```

Weiters erhält man beobachtete und erwartete Häufigkeiten nach Patterns, sowie die entsprechenden EAP-Schätzer in Tabellenform.

```
OBSERVED(EXPECTED)    STD.  :     EAP (S.D.)  :  PATTERN
                      RES.  :                 :

    28.0(    28.0)    0.00  :   -1.01 ( 0.86) :  11
    15.0(    15.1)   -0.03  :   -0.46 ( 0.81) :  12
     8.0(     6.2)    0.70  :   -0.17 ( 0.84) :  13
     5.0(     6.7)   -0.65  :    0.08 ( 0.90) :  14
    35.0(    35.3)   -0.04  :   -0.64 ( 0.81) :  21
    23.0(    24.7)   -0.34  :   -0.17 ( 0.77) :  22
    12.0(    11.5)    0.15  :    0.10 ( 0.79) :  23
    15.0(    13.4)    0.43  :    0.36 ( 0.84) :  24
    43.0(    40.0)    0.47  :   -0.23 ( 0.83) :  31
    35.0(    36.9)   -0.32  :    0.20 ( 0.78) :  32
    19.0(    20.5)   -0.33  :    0.49 ( 0.78) :  33
    29.0(    28.5)    0.09  :    0.79 ( 0.83) :  34
     6.0(     8.9)   -0.98  :    0.10 ( 0.89) :  41
    14.0(    10.1)    1.23  :    0.54 ( 0.82) :  42
     6.0(     6.6)   -0.24  :    0.85 ( 0.81) :  43
    11.0(    11.4)   -0.13  :    1.24 ( 0.86) :  44
```

5.4.2.1 Grafiken

MULTILOG verwendet dasselbe Interface, das auch bei BILOG zum Einsatz kommt, d.h. man kann wieder verschiedene Plottypen über ein Menü auswählen. In Abbildung 5.9 sieht man beispielsweise einen Matrixplot beider Items mit den entsprechenden Category Characteristic Curves.

Natürlich kann man auch anhand der geschätzten Daten mit anderen Programmen (R, Stata, SPSS, etc.) geeignete Grafiken erzeugen. In Abbildung 5.10 sind beispielsweise die erwarteten Häufigkeiten der jeweiligen EAP-Schätzer und die entsprechenden Antwortmuster beider Items aufgetragen, wo-

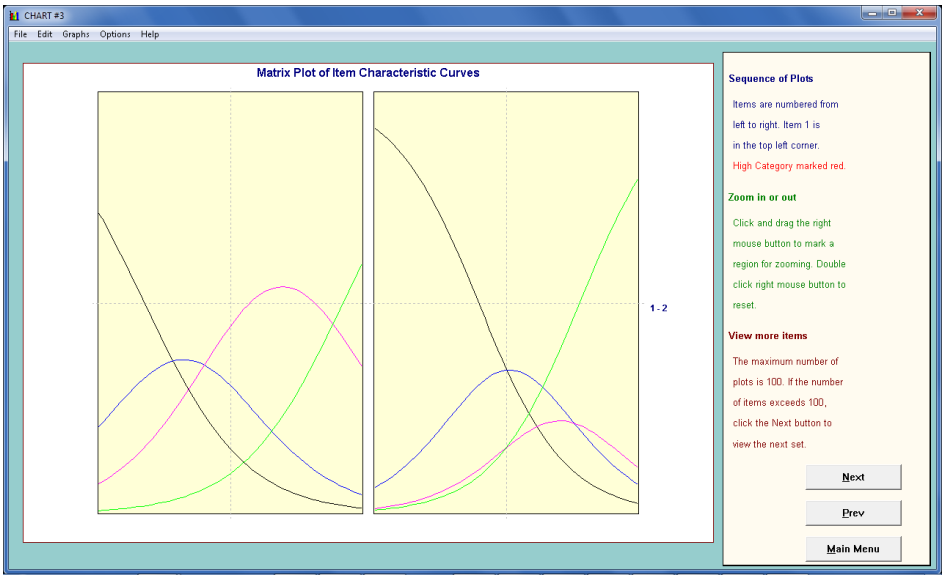


CHART #3
File Edit Graphs Options Help
Matrix Plot of Item Characteristic Curves
1 - 2
Sequence of Plots
Items are numbered from left to right. Item 1 is in the top left corner.
High Category marked red.
Zoom in or out
Click and drag the right mouse button to mark a region for zooming. Double click right mouse button to reset.
View more items
The maximum number of plots is 100. If the number of items exceeds 100, click the Next button to view the next set.
Next
Prev
Main Menu

Abb. 5.9 Matrix Plot der ICCs

durch man einerseits die Verteilung der Fähigkeitsparameter kennt und andererseits sieht, welche Antwortmuster wo am Kontinuum liegen.

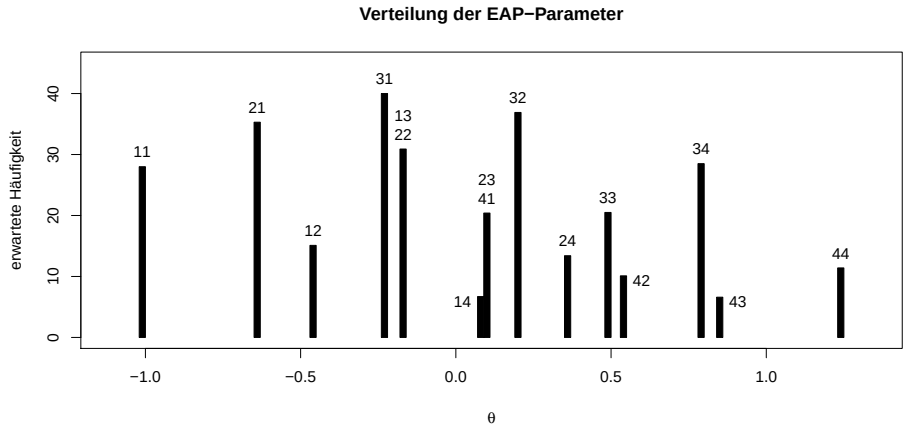


Abb. 5.10 Erwartete Verteilung der EAP-Parameter

5.5 Diskussion

Die beiden Programme haben viel dazu beigetragen, Item Response Modelle zugänglich und durchführbar zu machen. Vor allem in der Vergangenheit boten sie interessierten Forschern die Möglichkeit, IRT ohne übermäßigen technischen Hintergrund anzuwenden, wobei Einstiegshürden durch entsprechende Wizards gesenkt wurden. In diesem Beitrag konnte nur an der Oberfläche der Programme gekratzt werden, da insbesondere MULTILOG durch entsprechende Designmatrizen viele unterschiedliche Modelle schätzen kann, die auf den ersten Blick nicht ins Auge fallen. Auch die mitgelieferte Dokumentation ist umfangreich und mit vielen Beiträgen namhafter Personen versehen.

Bei allen Vorzügen dieses ‚altehrwürdigen' Programms darf man jedoch nicht die Nachteile der langen Tradition außer Acht lassen. Die Dialoge, die an sich helfen sollen, stellen vor allem Anfängern die

eine oder andere Falle, professionelle druckreife Grafiken, wie man sie von anderen Softwarepaketen gewohnt ist, sind den Paketen nur schwierig zu entlocken und die zuvor erwähnte Flexibilität erfordert die Einarbeitung in eine weitere ‚Programmiersprache'. Es ist daher fraglich, wie viel des Potentials der Programme tatsächlich zum Einsatz kommt, da sich Gelegenheits-IRT-User wahrscheinlich nicht durch die Matrizenspezifikationen kämpfen, um alle Modelle schätzen zu können. Dennoch sind BILOG und MULTILOG aufgrund ihrer Flexibilität und der Vielzahl an unterschiedlichen Modellen, Schätzmethoden etc. auch heute noch wichtige Pakete im IRT-Sektor.

Kapitel 6
Extended Rasch Modeling: The R Package eRm

Patrick Mair und Reinhold Hatzinger

Zusammenfassung This package vignette is an update of the eRm papers by published in a special issue on Psychometrics in the Journal of Statistical Software and in Psychology Science (Mair und Hatzinger, 2007a,b). Since the publication of these papers various extensions and additional features have been incorporated into the package. We start with a methodological introduction to extended Rasch models followed by a general program description and application topics. The package allows for the computation of simple Rasch models, rating scale models, partial credit models and linear extensions of these. The incorporation of such linear structures allows for modeling the effects of covariates and enables the analysis of repeated categorical measurements. The item parameter estimation is performed by means of CML, for the person parameters we use ordinary ML. The estimation routines work for incomplete data matrices as well. Based on these estimators, item-wise and global goodness-of-fit statistics are described and various plots are presented.

6.1 Introduction

Rost (1999) claimed in his article that "even though the Rasch model has been existing for such a long time, 95% of the current tests in psychology are still constructed by using methods from classical test theory"(p. 140). Basically, he quotes the following reasons why the Rasch model (RM) is being rarely used: The Rasch model in its original form (Rasch, 1960), which was limited to dichotomous items, is arguably too restrictive for practical testing purposes. Thus, researchers should focus on extended Rasch models. In addition, Rost argues that there is a lack of user-friendly software for the computation of such models. Hence, there is a need for a comprehensive, user-friendly software routine. Corresponding recent discussions can be found in Kubinger (2005) and Borsboom (2006).

In addition to the RM, the models that can be computed by means of the eRm package are: the linear logistic test model (Scheiblechner, 1972), the rating scale model (Andrich, 1978b), the linear rating scale model (Fischer und Parzer, 1991), the partial credit model (Masters, 1982), and the linear partial credit model (Glas und Verhelst, 1989; Fischer und Ponocny, 1994). These models and their main characteristics are presented in Section 6.2.

Concerning parameter estimation, these models have an important feature in common: Separability of item and person parameters. This implies that the item parameters β can be estimated without estimating the person parameters achieved by conditioning the likelihood on the sufficient person raw score. This conditional maximum likelihood (CML) approach is described in Section 6.3.

Several diagnostic tools and tests to evaluate model fit are presented in Section 6.4.

In Section 6.5, the corresponding implementation in R (R Development Core Team, 2007) is described by means of several examples. The eRm package uses a design matrix approach which allows to reparameterize the item parameters to model common characteristics of the items or to enable the

Ursprünglich erschienen in: Mair und Hatzinger (2007a)
WU Wien, Institut für Statistik und Mathematik • patrick.mair@wu.ac.at
WU Wien, Institut für Statistik und Mathematik • reinhold.hatzinger@wu.ac.at

user to impose repeated measurement designs as well as group contrasts. By combining these types of contrasts one allows that the item parameter may differ over time with respect to certain subgroups. To illustrate the flexibility of the `eRm` package some examples are given to show how suitable design matrices can be constructed.

6.2 Extended Rasch models

6.2.1 General expressions

Briefly after the first publication of the basic Rasch Model (Rasch, 1960), the author worked on polytomous generalizations which can be found in Rasch (1961). Andersen (1995) derived the representations below which are based on Rasch's general expression for polytomous data. The data matrix is denoted as $\mathbf{X}$ with the persons in the rows and the items in the columns. In total there are $v = 1,...,n$ persons and $i = 1,...,k$ items. A single element in the data matrix $\mathbf{X}$ is indexed by x_{vi}. Furthermore, each item I_i has a certain number of response categories, denoted by $h = 0,...,m_i$. The corresponding probability of response h on item i can be derived in terms of the following two expressions (Andersen, 1995):

$$P(X_{vi} = h) = \frac{\exp[\phi_h(\theta_v + \beta_i) + \omega_h]}{\sum_{l=0}^{m_i} \exp[\phi_l(\theta_v + \beta_i) + \omega_l]} \tag{6.1}$$

or

$$P(X_{vi} = h) = \frac{\exp[\phi_h\theta_v + \beta_{ih}]}{\sum_{l=0}^{m_i} \exp[\phi_l\theta_v + \beta_{il}]}. \tag{6.2}$$

Here, ϕ_h are scoring functions for the item parameters, θ_v are the uni-dimensional person parameters, and β_i are the item parameters. In Equation 6.1, ω_h corresponds to category parameters, whereas in Equation 6.2 β_{ih} are the item-category parameters. The meaning of these parameters will be discussed in detail below. Within the framework of these two equations, numerous models have been suggested that retain the basic properties of the Rasch model so that CML estimation can be applied.

6.2.2 Representation of extended Rasch models

For the ordinary Rasch model for dichotomous items, Equation 6.1 reduces to

$$P(X_{vi} = 1) = \frac{\exp(\theta_v - \beta_i)}{1 + \exp(\theta_v - \beta_i)}. \tag{6.3}$$

The main assumptions, which hold as well for the generalizations presented in this paper, are: unidimensionality of the latent trait, sufficiency of the raw score, local independence, and parallel item characteristic curves (ICCs). Corresponding explanations can be found, e.g., in Fischer (1974) and mathematical derivations and proofs in Fischer (1995a).

For dichotomous items, Scheiblechner (1972) proposed the (even more restricted) linear logistic test model (LLTM), later formalized by Fischer (1973), by splitting up the item parameters into the linear combination

$$\beta_i = \sum_{j=1}^{p} w_{ij}\eta_j. \tag{6.4}$$

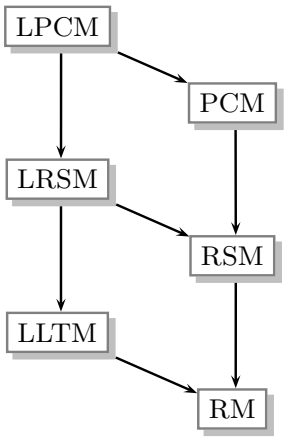


Abb. 6.1 Model hierarchy

Scheiblechner (1972) explained the dissolving process of items in a test for logics ("Mengenrechentest") by so-called "cognitive operations"η_j such as negation, disjunction, conjunction, sequence, intermediate result, permutation, and material. Note that the weights w_{ij} for item i and operation j have to be fixed a priori. Further elaborations about the cognitive operations can be found in Fischer (1974, p. 361ff.). Thus, from this perspective the LLTM is more parsimonous than the Rasch model.

Though, there exists another way to look at the LLTM: A generalization of the basic Rasch model in terms of repeated measures and group contrasts. It should be noted that both types of reparameterization also apply to the linear rating scale model (LRSM) and the linear partial credit model (LPCM) with respect to the basic rating scale model (RSM) and the partial credit model (PCM) presented below. Concerning the LLTM, the possibility to use it as a generalization of the Rasch model for repeated measurements was already introduced by Fischer (1974). Over the intervening years this suggestion has been further elaborated. Fischer (1995b) discussed certain design matrices which will be presented in Section 6.2.3 and on the basis of examples in Section 6.5.

At this point we will focus on a simple polytomous generalization of the Rasch model, the RSM (Andrich, 1978b), where each item I_i must have the same number of categories. Pertaining to Equation 6.1, ϕ_h may be set to h with $h = 0,...,m$. Since in the RSM the number of item categories is constant, m is used instead of m_i. Hence, it follows that

$$P(X_{vi} = h) = \frac{\exp[h(\theta_v + \beta_i) + \omega_h]}{\sum_{l=0}^{m} \exp[l(\theta_v + \beta_i) + \omega_l]}, \tag{6.5}$$

with k item parameters $\beta_1,...,\beta_k$ and $m+1$ category parameters $\omega_0,...,\omega_m$. This parameterization causes a scoring of the response categories C_h which is constant over the single items. Again, the item parameters can be split up in a linear combination as in Equation 6.4. This leads to the LRSM proposed by Fischer und Parzer (1991).

Finally, the PCM developed by Masters (1982) and its linear extension, the LPCM (Fischer und Ponocny, 1994), are presented. The PCM assigns one parameter β_{ih} to each $I_i \times C_h$ combination for $h = 0,...,m_i$. Thus, the constant scoring property must not hold over the items and in addition, the items can have different numbers of response categories denoted by m_i. Therefore, the PCM can be regarded as a generalization of the RSM and the probability for a response of person v on category h (item i) is defined as

$$P(X_{vih} = 1) = \frac{\exp[h\theta_v + \beta_{ih}]}{\sum_{l=0}^{m_i} \exp[l\theta_v + \beta_{il}]}. \tag{6.6}$$

It is obvious that (6.6) is a simplification of (6.2) in terms of $\phi_h = h$. As for the LLTM and the LRSM, the LPCM is defined by reparameterizing the item parameters of the basic model, i.e.,

$$\beta_{ih} = \sum_{j=1}^{p} w_{ihj}\eta_j. \tag{6.7}$$

These six models constitute a hierarchical order as displayed in Figure 6.1. This hierarchy is the base for a unified CML approach presented in the next section. It is outlined again that the linear extension models can be regarded either as generalizations or as more restrictive formulations pertaining to the underlying base model. The hierarchy for the basic model is straightforward: The RM allows only items with two categories, thus each item is represented by one parameter β_i. The RSM allows for more than two (ordinal) categories each represented by a category parameter ω_h. Due to identifiability issues, ω_0 and ω_1 are restricted to 0. Hence, the RM can be seen as a special case of the RSM whereas, the RSM in turn, is a special case of the PCM. The latter model assigns the parameter β_{ih} to each $I_i \times C_h$ combination.

To conclude, the most general model is the LPCM. All other models can be considered as simplifications of Equation 6.6 combined with Equation 6.7. As a consequence, once an estimation procedure is established for the LPCM, this approach can be used for any of the remaining models. This is what we quote as *unified CML approach*. The corresponding likelihood equations follow in Section 6.3.

6.2.3 The concept of virtual items

When operating with longitudinal models, the main research question is whether an individual's test performance changes over time. The most intuitive way would be to look at the shift in ability θ_v across time points. Such models are presented e.g. in Mislevy (1985), Glas (1992), and discussed by Hoijtink (1995).

Yet there exists another look onto time dependent changes, as presented in Fischer (1995b, p 158ff.): The person parameters are fixed over time and instead of them the item parameters change. The basic idea is that one item I_i is presented at two different times to the same person S_v is regarded as a pair of *virtual items*. Within the framework of extended Rasch models, any change in θ_v occuring between the testing occasions can be described without loss of generality as a change of the item parameters, instead of describing change in terms of the person parameter. Thus, with only two measurement points, I_i with the corresponding parameter β_i generates two virtual items I_r and I_s with associated item parameters β_r^* and β_s^*. For the first measurement point $\beta_r^* = \beta_i$, whereas for the second $\beta_s^* = \beta_i + \tau$. In this linear combination the β^*-parameters are composed additively by means of the real item parameters β and the treatment effects τ. This concept extends to an arbitrary number of time points or testing occasions.

Correspondingly, for each measurement point t we have a vector of *virtual item parameters* $\boldsymbol{\beta}^{*(t)}$ of length k. These are linear reparameterizations of the original $\boldsymbol{\beta}^{(t)}$, and thus the CML approach can be used for estimation. In general, for a simple LLTM with two measurement points the design matrix $\boldsymbol{W}$ is of the form as given in Table 6.1.

		η_1	η_2	$\cdots$	η_k	η_{k+1}
Time 1	$\beta_1^{*(1)}$	1	0	0	0	0
	$\beta_2^{*(1)}$	0	1	0	0	0
	$\vdots$			$\ddots$		$\vdots$
	$\beta_k^{*(1)}$	1	0	0	1	0
Time 2	$\beta_{k+1}^{*(2)}$	1	0	0	0	1
	$\beta_{k+2}^{*(2)}$	0	1	0	0	1
	$\vdots$			$\ddots$		$\vdots$
	$\beta_{2k}^{*(2)}$	1	0	0	1	1

Tabelle 6.1 A design matrix for an LLTM with two timepoints.

The parameter vector $\beta^{*(1)}$ represents the item parameters for the first test occasion, $\beta^{*(2)}$ the parameters for the second occasion. It might be of interest whether these vectors differ. The corresponding trend contrast is η_{k+1}. Due to this contrast, the number of original β-parameters is doubled by introducing the $2k$ virtual item parameters. If we assume a constant shift for all item parameters, it is only necessary to estimate $\hat{\eta}' = (\hat{\eta}_1, ..., \hat{\eta}_{k+1})$ where $\hat{\eta}_{k+1}$ gives the amount of shift. Since according to (6.4), the vector $\hat{\beta}^*$ is just a linear combination of $\hat{\eta}$.

As mentioned in the former section, when using models with linear extensions it is possible to impose group contrasts. By doing this, one allows that the item difficulties are different across subgroups. However, this is possible only for models with repeated measurements and virtual items since otherwise the introduction of a group contrast leads to overparameterization and the group effect cannot be estimated by using CML.

Table 6.2 gives an example for a repeated measurement design where the effect of a treatment is to be evaluated by comparing item difficulties regarding a control and a treatment group. The number of virtual parameters is doubled compared to the model matrix given in Table 6.1.

			η_1	η_2	$\cdots$	η_k	η_{k+1}	η_{k+2}
Time 1	Group 1	$\beta_1^{*(1)}$	1	0	0	0	0	0
		$\beta_2^{*(1)}$	0	1	0	0	0	0
		$\vdots$			$\ddots$		$\vdots$	$\vdots$
		$\beta_k^{*(1)}$	1	0	0	1	0	0
	Group 2	$\beta_{k+1}^{*(1)}$	1	0	0	0	0	0
		$\beta_{k+2}^{*(1)}$	0	1	0	0	0	0
		$\vdots$			$\ddots$		$\vdots$	$\vdots$
		$\beta_{2k}^{*(1)}$	1	0	0	1	0	0
Time 2	Group 1	$\beta_1^{*(2)}$	1	0	0	0	1	0
		$\beta_2^{*(2)}$	0	1	0	0	1	0
		$\vdots$			$\ddots$		$\vdots$	$\vdots$
		$\beta_k^{*(2)}$	1	0	0	1	1	0
	Group 2	$\beta_{k+1}^{*(2)}$	1	0	0	0	1	1
		$\beta_{k+2}^{*(2)}$	0	1	0	0	1	1
		$\vdots$			$\ddots$		$\vdots$	$\vdots$
		$\beta_{2k}^{*(2)}$	1	0	0	1	1	1

Tabelle 6.2 Design matrix for a repeated measurements design with treatment and control group.

Again, η_{k+1} is the parameter that refers to the time contrast, and η_{k+2} is a group effect within measurement point 2. More examples are given in Section 6.5 and further explanations can be found in Fischer (1995b), Fischer und Ponocny (1994), and in the software manual for the LPCM-Win program by Fischer und Ponocny-Seliger (1998).

By introducing the concept of virtual persons, eRm allows for the computation of the linear logistic test model with relaxed assumptions (LLRA Fischer, 1977). Corresponding explanations will be given in a subsequent version of this vignette.

6.3 Estimation of item and person parameters

6.3.1 CML for item parameter estimation

The main idea behind the CML estimation is that the person's raw score $r_v = \sum_{i=1}^{k} x_{vi}$ is a sufficient statistic. Thus, by conditioning the likelihood onto $\boldsymbol{r}' = (r_1, ..., r_n)$, the person parameters $\boldsymbol{\theta}$, which in this context are nuisance parameters, vanish from the likelihood equation, thus, leading to consistently estimated item parameters $\hat{\boldsymbol{\beta}}$.

Some restrictions have to be imposed on the parameters to ensure identifiability. This can be achieved, e.g., by setting certain parameters to zero depending on the model. In the Rasch model one item parameter has to be fixed to 0. This parameter may be considered as baseline difficulty. In addition, in the RSM the category parameters ω_0 and ω_1 are also constrained to 0. In the PCM all parameters representing the first category, i.e. β_{i0} with $i = 1, \ldots, k$, and one additional item-category parameter, e.g., β_{11} have to be fixed. For the linear extensions it holds that the β-parameters that are fixed within a certain condition (e.g. first measurement point, control group etc.) are also constrained in the other conditions (e.g. second measurement point, treatment group etc.).

At this point, for the LPCM the likelihood equations with corresponding first and second order derivatives are presented (i.e. *unified CML equations*). In the first version of the `eRm` package numerical approximations of the Hessian matrix are used. However, to ensure numerical accuracy and to speed up the estimation process, it is planned to implement the analytical solution as given below.

The conditional log-likelihood equation for the LPCM is

$$\log L_c = \sum_{i=1}^{k} \sum_{h=1}^{m_i} x_{+ih} \sum_{j=1}^{p} w_{ihj} \eta_j - \sum_{r=1}^{r_{max}} n_r \log \gamma_r. \tag{6.8}$$

The maximal raw score is denoted by r_{max} whereas the number of subjects with the same raw score is quoted as n_r. Alternatively, by going down to an individual level, the last sum over r can be replaced by $\sum_{v=1}^{n} \log \gamma_{r_v}$. It is straightforward to show that the LPCM as well as the other extended Rasch models, define an exponential family (Andersen, 1983). Thus, the raw score r_v is minimally sufficient for θ_v and the item totals $x_{\cdot ih}$ are minimally sufficient for β_{ih}.

Crucial expressions are the γ-terms which are known as *elementary symmetric functions*. More details about these terms are given in the next section. However, in the `eRm` package the numerically stable *summation algorithm* as suggested by Andersen (1972) is implemented. Fischer und Ponocny (1994) adopted this algorithm for the LPCM and devised also the first order derivative for computing the corresponding derivative of $\log L_c$:

$$\frac{\partial \log L_c}{\partial \eta_a} = \sum_{i=1}^{k} \sum_{h=1}^{m_i} w_{iha} \left(x_{+ih} - \epsilon_{ih} \sum_{r=1}^{r_{max}} n_r \frac{\gamma_r^{(i)}}{\gamma_r} \right). \tag{6.9}$$

It is important to mention that for the CML-representation, the multiplicative Rasch expression is used throughout equations 6.1 to 6.7, i.e., $\epsilon_i = \exp(-\beta_i)$ for the person parameter. Therefore, ϵ_{ih} corresponds to the reparameterized item × category parameter whereas $\epsilon_{ih} > 0$. Furthermore, $\gamma_r^{(i)}$ are the first order derivatives of the γ-functions with respect to item i. The index a in η_a denotes the first derivative with respect to the a^{th} parameter.

For the second order derivative of $\log L_c$, two cases have to be distinguished: the derivatives for the off-diagonal elements and the derivatives for the main diagonal elements. The item categories with respect to the item index i are coded with h_i, and those referring to item l with h_l. The second order derivatives of the γ-functions with respect to items i and l are denoted by $\gamma_r^{(i,l)}$. The corresponding likelihood expressions are

$$\frac{\partial \log L_c}{\partial \eta_a \eta_b} = -\sum_{i=1}^{k}\sum_{h_i=1}^{m_i} w_{ih_ia} w_{ih_ib} \epsilon_{ih_i} \sum_{r=1}^{r_{max}} n_r \frac{\log \gamma_{r-h_i}}{\gamma_r} \tag{6.10}$$
$$-\sum_{i=1}^{k}\sum_{h_i=1}^{m_i}\sum_{l=1}^{k}\sum_{h_l=1}^{m_l} w_{ih_ia} w_{lh_lb} \left[\epsilon_{ih_i}\epsilon_{lh_l} \left(\sum_{r=1}^{r_{max}} n_r \frac{\gamma_r^{(i)}\gamma_r^{(l)}}{\gamma_r^2} - \sum_{r=1}^{r_{max}} n_r \frac{\gamma_r^{(i,l)}}{\gamma_r} \right) \right]$$

for $a \neq b$, and

$$\frac{\partial \log L_c}{\partial \eta_a^2} = -\sum_{i=1}^{k}\sum_{h_i=1}^{m_i} w_{ih_ia}^2 \epsilon_{ih_i} \sum_{r=1}^{r_{max}} n_r \frac{\log \gamma_{r-h_i}}{\gamma_r} \tag{6.11}$$
$$-\sum_{i=1}^{k}\sum_{h_i=1}^{m_i}\sum_{l=1}^{k}\sum_{h_l=1}^{m_l} w_{ih_ia} w_{lh_la} \epsilon_{ih_i}\epsilon_{lh_l} \sum_{r=1}^{r_{max}} n_r \frac{\gamma_{r-h_i}^{(i)}\gamma_{r-h_l}^{(l)}}{\gamma_r^2}$$

for $a = b$.

To solve the likelihood equations with respect to $\hat{\eta}$, a Newton-Raphson algorithm is applied. The update within each iteration step s is performed by

$$\hat{\eta}_s = \hat{\eta}_{s-1} - \mathbf{H}_{s-1}^{-1}\delta_{s-1}. \tag{6.12}$$

The starting values are $\hat{\eta}_0 = 0$. $\mathbf{H}_{s-1}^{-1}$ is the inverse of the Hessian matrix composed by the elements given in Equation 6.10 and 6.11 and δ_{s-1} is the gradient at iteration $s-1$ as specified in Equation 6.9. The iteration stops if the likelihood difference $\left|\log L_c^{(s)} - \log L_c^{(s-1)}\right| \leq \varphi$ where φ is a predefined (small) iteration limit. Note that in the current version (0.13-0) $\mathbf{H}$ is approximated numerically by using the `nlm` Newton-type algorithm provided in the `stats` package. The analytical solution as given in Equation 6.10 and 6.11 will be implemented in the subsequent version of `eRm`.

6.3.2 Mathematical properties of the CML estimates

A variety of estimation approaches for IRT models in general and for the Rasch model in particular are available: The *joint maximum likelihood* (JML) estimation as proposed by Wright und Panchapakesan (1969) which is not recommended since the estimates are not consistent (see e.g. Haberman, 1977). The basic reason for that is that the person parameters θ are nuisance parameters; the larger the sample size, the larger the number of parameters.

A well-known alternative is the *marginal maximum likelihood* (MML) estimation (Bock und Aitkin, 1981): A distribution $g(\theta)$ for the person parameters is assumed and the resulting situation corresponds to a mixed-effects ANOVA: Item difficulties can be regarded as fixed effects and person abilities as random effects. Thus, IRT models fit into the framework of *generalized linear mixed models* (GLMM) as elaborated in de Boeck und Wilson (2004). By integrating over the ability distribution the random nuisance parameters can be removed from the likelihood equations. This leads to consistent estimates of the item parameters. Further discussions of the MML approach with respect to the CML method will follow.

For the sake of completeness, some other methods for the estimation of the item parameters are the following: Anderson et al. (2007) propose a Pseudo-ML approach, Molenaar (1995) and Linacre (2004a) give an overview of various (heuristic) non-ML methods, Bayesian techniques can be found in Baker und Kim (2004, Chapter 7), and for nonparameteric approaches it is referred to de Leeuw und Verhelst (1986).

However, back to CML, the main idea behind this approach is the assumption that the raw score r_v is a minimal sufficient statistic for θ_v. Starting from the equivalent multiplicative expression of Equation 6.1 with $\xi_v = \exp(\theta_v)$ and $\epsilon_i = \exp(-\beta_i)$, i.e.,

$$P(X_{vi} = 1) = \frac{\xi_v \epsilon_i}{1 + \xi_v \epsilon_i}, \tag{6.13}$$

the following likelihood for the response pattern $\boldsymbol{x}_v$ for a certain subject v results:

$$P(\boldsymbol{x}_v | \xi_v, \boldsymbol{\epsilon}) = \prod_{i=1}^{k} \frac{(\xi_v \epsilon_i)^{x_{vi}}}{1 + \xi_v \epsilon_i} = \frac{{\theta_v}^{r_v} \prod_{i=1}^{k} {\epsilon_i}^{x_{vi}}}{\prod_{i=1}^{k} (1 + \xi_v \epsilon_i)}. \tag{6.14}$$

Using the notation $\boldsymbol{y} = (y_1, \ldots, y_k)$ for all possible response patterns with $\sum_{i=1}^{k} y_i = r_v$, the probability for a fixed raw score r_v is

$$P(r_v | \xi_v, \boldsymbol{\epsilon}) = \sum_{\boldsymbol{y}|r_v} \prod_{i=1}^{k} \frac{(\xi_v \epsilon_i)^{x_{vi}}}{1 + \xi_v \epsilon_i} = \frac{{\theta_v}^{r_v} \sum_{\boldsymbol{y}|r_v} \prod_{i=1}^{k} {\epsilon_i}^{x_{vi}}}{\prod_{i=1}^{k} (1 + \xi_v \epsilon_i)}. \tag{6.15}$$

The crucial term with respect to numerical solutions of the likelihood equations is the second term in the numerator:

$$\gamma_r(\epsilon_i) \equiv \sum_{\boldsymbol{y}|r_v} \prod_{i=1}^{k} {\epsilon_i}^{x_{vi}} \tag{6.16}$$

These are the *elementary symmetric functions* (of order r). An overview of efficient computational algorithms and corresponding simulation studies can be found in Liou (1994). The `eRm` package uses the summation algorithm as proposed by Andersen (1972).

Finally, by collecting the different raw scores into the vector $\boldsymbol{r}$ the conditional probability of observing response pattern $\boldsymbol{x}_v$ with given raw score r_v is

$$P(\boldsymbol{x}_v | r_v, \boldsymbol{\epsilon}) = \frac{P(\boldsymbol{x}_v | \xi_v, \boldsymbol{\epsilon})}{P(r_v | \xi_v, \boldsymbol{\epsilon})}. \tag{6.17}$$

By taking the product over the persons (independence assumption), the (conditional) likelihood expression for the whole sample becomes

$$L(\boldsymbol{\epsilon} | \boldsymbol{r}) = P(\boldsymbol{x} | \boldsymbol{r}, \boldsymbol{\epsilon}) = \prod_{v=1}^{n} \frac{\prod_{i=1}^{k} {\epsilon_i}^{x_{vi}}}{\gamma_{r_v}}. \tag{6.18}$$

With respect to raw score frequencies n_r and by reintroducing the β-parameters, (6.18) can be reformulated as

$$L(\boldsymbol{\beta} | \boldsymbol{r}) = \frac{\exp\left(\sum_{i=1}^{k} x_{+i} \beta_i\right)}{\prod_{r=0}^{k} \gamma_r^{n_r}}, \tag{6.19}$$

where x_{+i} are the item raw scores. It is obvious that by conditioning the likelihood on the raw scores $\boldsymbol{r}$, the person parameters completely vanished from the expression. As a consequence, the parameters $\hat{\boldsymbol{\beta}}$ can be estimated without knowledge of the subject's abilities. This issue is referred as *person-free item assessment* and we will discuss this topic within the context of specific objectivity in the next section.

Pertaining to asymptotical issues, it can be shown that under mild regularity conditions (Pfanzagl, 1994) the CML estimates are consistent for $n \to \infty$ and k fixed, unbiased, asymptotically efficient, and normally distributed (Andersen, 1970). For the computation of a Rasch model, comparatively small samples are sufficient to get reliable estimates (Fischer, 1988). Whether the MML estimates are unbiased depends on the correct specification of the ability distribution $g(\theta)$. In case of an incorrect assumption, the estimates are biased which is surely a drawback of this method. If $g(\theta)$ is specified appropriately, the CML and MML estimates are asymptotically equivalent (Pfanzagl, 1994).

Fischer (1981) elaborates on the conditions for the existence and the uniqueness of the CML estimates. The crucial condition for the data matrix is that $\boldsymbol{X}$ has to be *well-conditioned*. To introduce this issue it is convenient to look at a matrix which is *ill-conditioned*: A matrix is ill-conditioned if there exists a partition of the items into two nonempty subsets such that all of a group of subjects

responded correctly to items $i+1,\ldots,k$ (X_2) and all of all other subjects failed for items $1,\ldots,i$ (X_3), i.e.,

$$X = \left(\begin{array}{c|c} X_1 & X_2 \\ \hline X_3 & X_4 \end{array}\right) = \left(\begin{array}{c|c} & \begin{matrix} 1 & \ldots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \ldots & 1 \end{matrix} \\ X_1 & \\ \hline \begin{matrix} 0 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 0 \end{matrix} & \\ & X_4 \end{array}\right)$$

Thus, following the definition in Fischer (1981): X will be called *well-conditioned* iff in every possible partition of the items into two nonempty subsets some subjects has given response 1 on some item in the first set and response 0 on some item in the second set. In this case a unique solution for the CML estimates $\hat{\beta}$ exists.

This issue is important for structurally incomplete designs which often occur in practice; different subsets of items are presented to different groups of persons $g = 1,\ldots,G$ where $G \leq n$. As a consequence, the likelihood values have to be computed for each group separately and the joint likelihood is the product over the single group likelihoods. Hence, the likelihood in Equation 6.19 becomes

$$L(\beta|\boldsymbol{r}) = \prod_{g=1}^{G} \frac{\exp\left(\sum_{i=1}^{k} x_{+i}\beta_i\right)}{\prod_{r=0}^{k} \gamma_{g,r}{}^{n_{g,r}}} \tag{6.20}$$

This also implies the necessity to compute the elementary symmetric functions separately for each group. The `eRm` package can handle such structurally incomplete designs.

From the elaborations above it is obvious that from an asymptotical point of view the CML estimates are at least as good as the MML estimates. In the past, computational problems (speed, numerical accuracy) involved in calculating the elementary symmetric functions limited the practical usage of the CML approach (see e.g. Gustafsson, 1980). Nowadays, these issues are less crucial due to increased computer power.

In some cases MML estimation has advantages not shared by CML: MML leads to finite person parameters even for persons with zero and perfect raw score, and such persons are not removed from the estimation process (Molenaar, 1995). On he other hand the consideration of such persons does not seem meaningful from a substantial point of view since the person parameters are not reliable anymore – for such subjects the test is too difficult or too easy, respectively. Thus, due to these covering effects, a corresponding ability estimation is not feasible. However, if the research goal is to find ability distributions such persons should be regarded and MML can handle this.

When estimates for the person parameters are of interest some care has to be taken if the CML method is used since person parameters cancel from the estimation equations. Usually, they are estimated (once having obtained values for the item parameters) by inserting $\hat{\beta}$ (or equivalently $\hat{\epsilon}$) into Equation 6.14 and solving with respect to θ. Alternatively, Bayesian procedures are applicable (Hoijtink und Boomsma, 1995). It is again pointed out that each person in the sample gets an own parameter even though limited by the number of different raw scores.

6.3.3 CML and specific objectivity

In general, the Rasch model can be regarded as a measurement model: Starting from the (nominally scaled) 0/1-data matrix X, the person raw scores r_v are on an ordinal level. They, in turn, are used to estimate the item parameters β which are on an interval scale provided that the Rasch model holds.

Thus, Rasch models allow for comparisons between objects on an interval level. Rasch reasoned on requirements to be fulfilled such that a specific proposition within this context can be regarded as "scientific". His conclusions were that a basic requirement is the "objectivity" of comparisons (Rasch, 1961). This claim contrasts assumptions met in *classical test theory* (CTT). A major advantage of the Rasch model over CTT models is the *sample independence* of the results. The relevant concepts in CTT are based on a linear model for the "true score"leading to some indices, often correlation coefficients, which in turn depend on the observed data. This is a major drawback in CTT. According to Fischer (1974), sample independence in IRT models has the following implications:

- The person-specific results (i.e., essentially θ) do not depend on the assignment of a person to a certain subject group nor on the selected test items from an item pool Ψ.
- Changes in the skills of a person on the latent trait can be determined independently from its base level and independently from the selected item subset $\psi \subset \Psi$.
- From both theoretical and practical perspective the requirement for representativeness of the sample is obsolete in terms of a true random selection process.

Based on these requirements for parameter comparisons, Rasch (1977) introduced the term *specific objectivity*: *objective* because any comparison of a pair of parameters is independent of any other parameters or comparisons; *specifically objective* because the comparison made was relative to some specified frame of reference (Andrich, 1988b). In other words, if specific objectivity holds, two persons v and w with corresponding parameters θ_v and θ_w, are comparable independently from the remaining persons in the sample and independently from the presented item subset ψ. In turn, for two items i and j with parameters β_i and β_j, the comparison of these items can be accomplished independently from the remaining items in Ψ and independently from the persons in the sample.

The latter is crucial since it reflects completely what is called sample independence. If we think not only of comparing β_i and β_j but rather to estimate these parameters, we achieve a point where specific objectivity requires a procedure which is able to provide estimates $\hat{\boldsymbol{\beta}}$ that do not depend on the sample. This implies that $\hat{\boldsymbol{\beta}}$ should be computable without the involvement of $\boldsymbol{\theta}$. CML estimation fulfills this requirement: By conditioning on the sufficient raw score vector $\boldsymbol{r}$, $\boldsymbol{\theta}$ disappears from the likelihood equation and $L(\boldsymbol{\beta}|\boldsymbol{r})$ can be solved without knowledge of $\boldsymbol{\theta}$. This issue is referred to as *separability of item and person parameters* (see e.g. Wright und Masters, 1982). Furthermore, separability implies that no specific distribution should be assumed neither for the person nor for the item parameters (Rost, 2000). MML estimation requires such assumptions. At this point it is clear that CML estimation is the only estimation method within the Rasch measurement context fulfilling the requirement of *person-free item calibration* and, thus, it maps the epistemological theory of specific objectivity to a statistical maximum likelihood framework. Note that strictly speaking any statistical result based on sample observations is sample-dependent because any result depends at least on the sample size (Fischer, 1987). The estimation of the item parameters is "sample-independent", a term indicating the fact that the actually obtained sample of a certain population is not of relevance for the statistical inference on these parameters (Kubinger, 1989, p. 23).

6.3.4 Estimation of person parameters

CML estimation for person parameters is not recommended due to computational issues. The `eRm` package provides two methods for this estimation. The first is ordinary ML where the CML-based item parameters are plugged into the joint ML equation. The likelihood is optimized with respect to $\boldsymbol{\theta}$. Andersen (1995) gives a general formulation of this ML estimate with $r_v = r$ and $\theta_v = \theta$:

$$r - \sum_{i=1}^{k} \sum_{h=1}^{m_i} \frac{h \exp(h\theta + \hat{\beta}_{ih})}{\sum_{l=0}^{m_i} \exp(h\theta_v + \hat{\beta}_{il})} = 0 \tag{6.21}$$

Warm (1989) proposed a weighted likelihood estimation (WLE) which is more accurate compared to ML. For the dichotomous Rasch model the expression to be solved with respect to $\boldsymbol{\theta}$ is

$$P(\theta_v|\boldsymbol{x}_v, \hat{\boldsymbol{\beta}}) \propto \frac{exp(r_v\theta_v)}{\prod_i(1+exp(\theta_v-\hat{\beta}_i)} \sum_i p_{vi}(1-p_{vi}) \tag{6.22}$$

Again, the item parameter vector $\hat{\boldsymbol{\beta}}$ is used from CML. This approach will implemented in a subsequent eRm version. Additional explanations and simulation studies regarding person parameter estimation can be found in Hoijtink und Boomsma (1995).

6.4 Testing extended Rasch models

Testing IRT models involves two parts: First, item- and person-wise statistics can be examined; in particular item-fit and person-fit statistics. Secondly, based on CML properties, various model tests can be derived (see Glas und Verhelst, 1995a,b).

6.4.1 Item-fit and person-fit statistics

Commonly in IRT, items and persons are excluded due to item-fit and person-fit statistics. Both are residual based measures: The observed data matrix **X** is compared with the model probability matrix **P**. Computing standardized residuals for all observations gives the $n \times k$ residual matrix **R**. The squared column sums correspond to item-fit statistics and the squared row sums to person-fit statistics both of which are χ^2-distributed with the corresponding degrees of freedom. Based on these quantities unweighted (*outfit*) and weighted (*infit*) mean-square statistics can also be used to evaluate item and person fit (see e.g. Wright und Masters, 1982).

6.4.2 A Wald test for item elimination

A helpful implication of CML estimates is that subsequent test statistics are readily obtained and model tests are easy to carry out. Basically, we have to distinguish between test on item level and global model tests.

On item level, sample independence reflects the property that by splitting up the sample in, e.g., two parts, the corresponding parameter vectors $\hat{\boldsymbol{\beta}}^{(1)}$ and $\hat{\boldsymbol{\beta}}^{(2)}$ should be the same. Thus, when we want to achieve Rasch model fit those items have to be eliminated from the test which differ in the subsamples. This important issue in test calibration can be examined, e.g., by using a graphical model test. Fischer und Scheiblechner (1970) propose a $N(0,1)$-distributed test statistic which compares the item parameters for two subgroups:

$$z = \frac{\beta_i^{(1)} - \beta_i^{(2)}}{\sqrt{Var_i^{(1)} - Var_i^{(2)}}} \tag{6.23}$$

The variance term in the denominator is based on Fisher's function of "information in the sample". However, as Glas und Verhelst (1995a) point out discussing their Wald-type test that this term can be extracted directly from the variance-covariance matrix of the CML estimates. This Wald approach is provided in eRm by means of the function `Waldtest()`.

6.4.3 Andersen's likelihood-ratio test

In the `eRm` package the likelihood ratio test statistic LR, initially proposed by Andersen (1973) is computed for the RM, the RSM, and the PCM. For the models with linear extensions, LR has to be computed separately for each measurement point and subgroup.

$$LR = 2\left(\sum_{g=1}^{G} \log L_c(\hat{\eta}_g; \boldsymbol{X}_g) - \log L_c(\hat{\eta}; \boldsymbol{X})\right) \tag{6.24}$$

The underlying principle of this test statistic is that of *subgroup homogeneity* in Rasch models: for arbitrary disjoint subgroups $g = 1,...,G$ the parameter estimates $\hat{\eta}_g$ have to be the same. LR is asymptotically χ^2-distributed with df equal to the number of parameters estimated in the subgroups minus the number of parameters in the total data set. For the sake of computational efficiency, the `eRm` package performs a person raw score median split into two subgroups. In addition, a graphical model test (Rasch, 1960) based on these estimates is produced by plotting $\hat{\beta}_1$ against $\hat{\beta}_2$. Thus, critical items (i.e. those fairly apart from the diagonal) can be identified and eliminated. Further elaborations and additional test statistics for polytomous Rasch models can be found, e.g., in Glas und Verhelst (1995a).

6.4.4 Nonparametric ("exact") Tests

Based on the package `RaschSampler` by Verhelst et al. (2007) several Rasch model tests as proposed by (Ponocny, 2001) are provided.

6.4.5 Martin-Löf Test

Applying the LR principle to subsets of items, Martin-Löf (1973, see Glas und Verhelst, 1995a) suggested a statistic to evaluate if two groups of items are homogeneous, i.e., to test the unidimensionality axiom.

6.5 The eRm package and application examples

The underlying idea of the `eRm` package is to provide a user-friendly flexible tool to compute extended Rasch models. This implies, amongst others, an automatic generation of the design matrix $\mathbf{W}$. However, in order to test specific hypotheses the user may specify $\mathbf{W}$ allowing the package to be flexible enough for computing IRT-models beyond their regular applications. In the following subsections, various examples are provided pertaining to different model and design matrix scenarios. Due to intelligibility matters, the artificial data sets are kept rather small. A detailed description in German of applications of various extendend Rasch models using the `eRm` package can be found in Poinstingl et al. (2007).

6.5.1 Structure of the eRm package

Embedding `eRm` into the flexible framework of R is a crucial benefit over existing stand-alone programs like WINMIRA (von Davier, 1998), LPCM-WIN (Fischer und Ponocny-Seliger, 1998), and others.

Another important issue in the development phase was that the package should be flexible enough to allow for CML compatible polytomous generalizations of the basic Rasch model such as the RSM

and the PCM. In addition, by introducing a design matrix concept linear extensions of these basic models should be applicable. This approach resulted in including the LLTM, the LRSM and the LPCM as the most general model into the eRm package. For the latter model the CML estimation was implemented which can be used for the remaining models as well. A corresponding graphical representation is given in Figure 6.2.

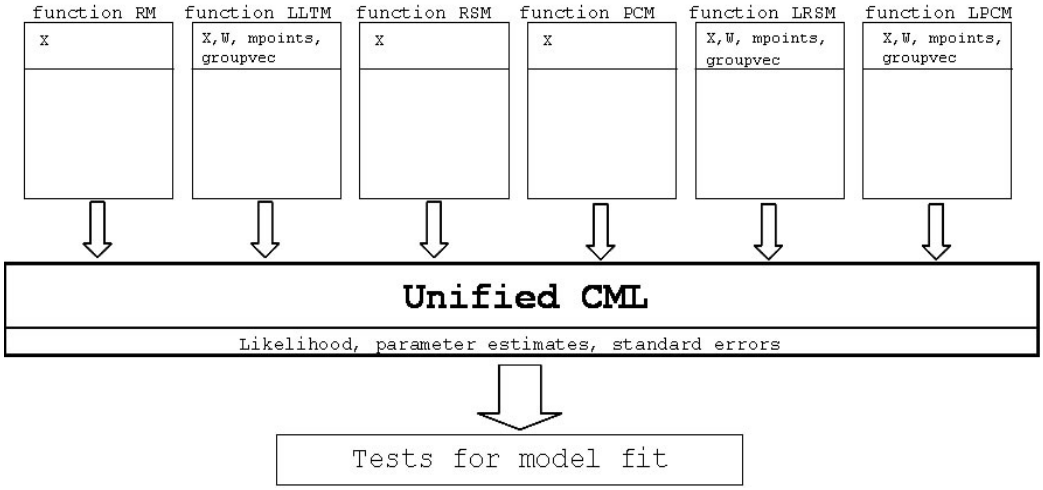


Abb. 6.2 Bodywork of the eRm routine

An important benefit of the package with respect to linearly extended models is that for certain models the design matrix W can be generated automatically (LPCM-WIN, Fischer und Ponocny-Seliger, 1998) also allows for specifying design matrices but in case of more complex models this can become a tedious task and the user must have a thorough understanding of establishing proper design structures). For repeated measurement models time contrasts in the eRm can be simply specified by defining the number of measurement points, i.e., `mpoints`. To regard group contrasts like, e.g., treatment and control groups, a corresponding vector (`groupvec`) can be specified that denotes which person belongs to which group. However, W can also be defined by the user.

A recently added feature of the routine is the option to allow for structurally missing values. This is required, e.g., in situations when different subsets of items are presented to different groups of subjects as described in Section 6.3.2. These person groups are identified automatically: In the data matrix X, those items which are not presented to a certain subject are declared as `NA`s, as usual in R.

After solving the CML equations by the Newton-Raphson method, the output of the routine consists of the "basic"parameter estimates $\hat{\eta}$, the corresponding variance-covariance matrix, and consequently the vector with the standard errors. Furthermore, the ordinary item parameter estimates $\hat{\beta}$ are computed by using the linear transformation $\hat{\beta} = W\hat{\eta}$. For ordinary Rasch models these basic parameters correspond to the item easiness. For the RM, the RSM, and the PCM, however, we display $\hat{\eta}$ as $-\hat{\eta}$, i.e., as difficulty. It has to be mentioned that the CML equation is solved with the restriction that one item parameter has to be fixed to zero (we use $\beta_1 = 0$). For the sake of interpretability, the resulting estimates $\hat{\beta}$ can easily be transformed into "sum-zero"restricted $\hat{\beta}^*$ by applying $\hat{\beta}_i^* = \hat{\beta}_i - \sum_i \hat{\beta}_i / k$. This transformation is also used for the graphical model test.

6.5.2 Example 1: Rasch model

We start the example section with a simple Rasch model based on a 100×30 data matrix. First, we estimate the item parameters using the function `RM()` and then the person parameters with `person.parameters()`.

```
> library(eRm)
> data(raschdat1)
> res.rasch <- RM(raschdat1)
> pres.rasch <- person.parameter(res.rasch)
```

Then we use Andersen's LR-test for goodness-of-fit with mean split criterion:

```
> lrres.rasch <- LRtest(res.rasch, splitcr = "mean", se = TRUE)
> lrres.rasch

Andersen LR-test:
LR-value: 30.288
Chi-square df: 29
p-value:  0.4
```

We see that the model fits and a graphical representation of this result (subset of items only) is given in Figure 6.3 by means of a goodness-of-fit plot with confidence ellipses.

```
> plotGOF(lrres.rasch, beta.subset = c(14, 5, 18, 7, 1), tlab = "item",
+     conf = list(ia = FALSE, col = "blue", lty = "dotted"))
```

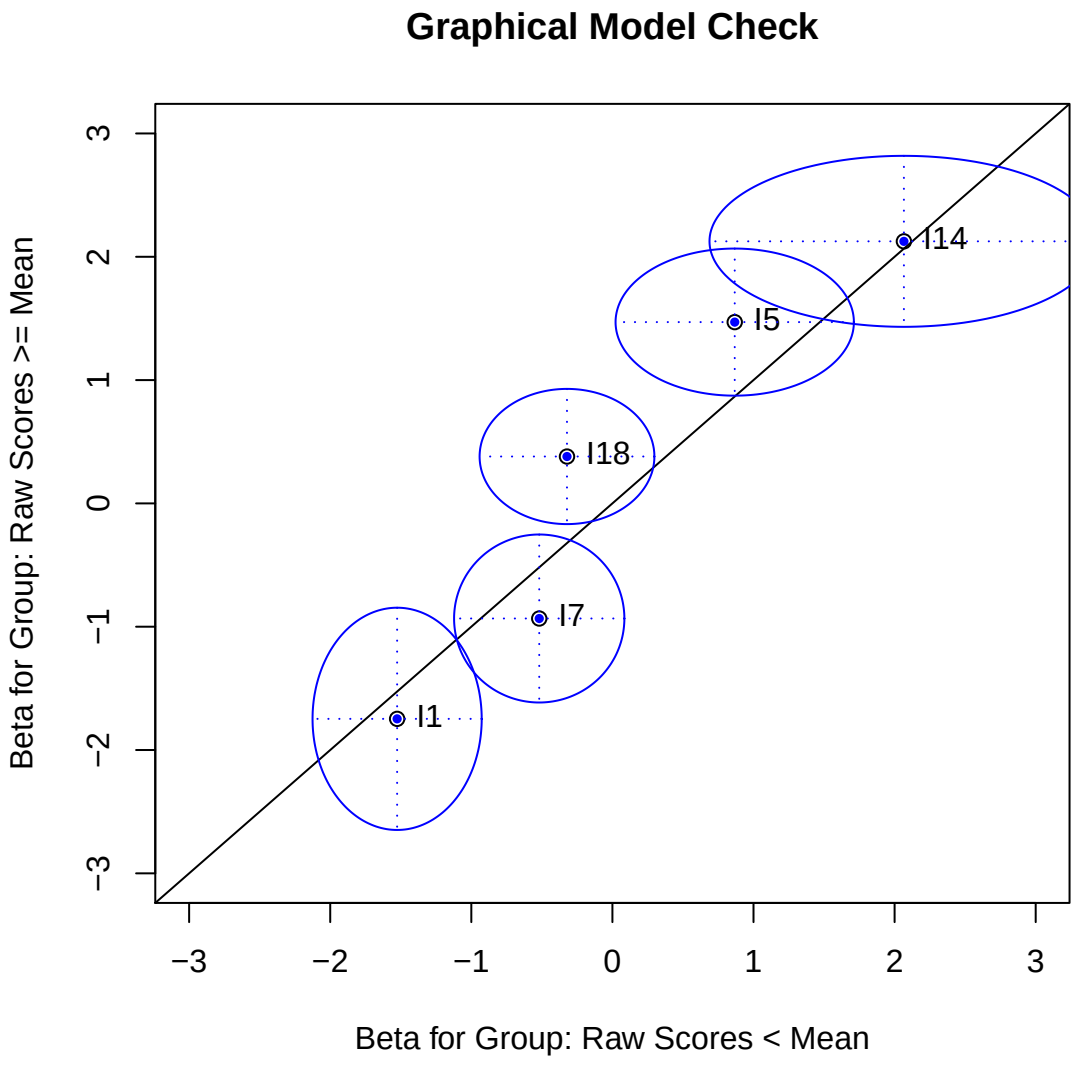


Abb. 6.3 Goodness-of-fit plot for some items with confidence ellipses.

To be able to draw confidence ellipses it is needed to set `se = TRUE` when computing the LR-test.

6.5.3 Example 2: LLTM as a restricted Rasch model

As mentioned in Section 6.2.2, also the models with the linear extensions on the item parameters can be seen as special cases of their underlying basic model. In fact, the LLTM as presented below and following the original idea by Scheiblechner (1972), is a restricted RM, i.e. the number of estimated parameters is smaller compared to a Rasch model. The data matrix **X** consists of $n = 15$ persons and $k = 5$ items. Furthermore, we specify a design matrix **W** (following Equation 6.4) with specific weight elements w_{ij}.

```
> data(lltmdat2)
> W <- matrix(c(1, 2, 1, 3, 2, 2, 2, 1, 1, 1), ncol = 2)
> res.lltm <- LLTM(lltmdat2, W)
> summary(res.lltm)

Results of LLTM estimation:

Call:  LLTM(X = lltmdat2, W = W)

Conditional log-likelihood: -31.65225
Number of iterations: 7
Number of parameters: 2

Basic Parameters eta with 0.95 CI:
      Estimate Std. Error lower CI upper CI
eta 1   -0.098      0.313   -0.711    0.516
eta 2    0.114      0.478   -0.823    1.051

Item Easiness Parameters (beta) with 0.95 CI:
        Estimate Std. Error lower CI upper CI
beta I1    0.130      1.098   -2.022    2.283
beta I2    0.033      1.302   -2.519    2.584
beta I3    0.016      0.651   -1.259    1.292
beta I4   -0.179      1.184   -2.500    2.141
beta I5   -0.081      0.903   -1.850    1.688
```

The `summary()` method provides point estimates and standard errors for the basic parameters and for the resulting item parameters. Note that item parameters in `eRm` are always estimated as easiness parameters according to equations 6.1 and 6.2 but not 6.3. If the sign is switched, the user gets difficulty parameters (the standard errors remain the same, of course). However, all plotting functions `plotGOF`, `plotICC`, `plotjointICC`, and `plotPImap`, as well as the function `thresholds` display the difficulty parameters. The same applies for the basic parameters η in the output of the RM, RSM, and PCM.

6.5.4 Example 3: RSM and PCM

Again, we provide an artificial data set now with $n = 300$ persons and $k = 4$ items; each of them with $m + 1 = 3$ categories. We start with the estimation of an RSM and, subsequently, we calculate the corresponding category-intersection parameters using the function `thresholds()`.

```
> data(pcmdat2)
> res.rsm <- RSM(pcmdat2)
> thresholds(res.rsm)

Design Matrix Block 1:
   Location Threshold 1 Threshold 2
```

```
I1  1.60712     0.59703     2.61721
I2  1.92251     0.91242     2.93260
I3  0.00331    -1.00678     1.01340
I4  0.50743    -0.50266     1.51752
```

The location parameter is basically the item difficulty and the thesholds are the points in the ICC plot given in Figure 6.4 where the category curves intersect:

```
> plotICC(res.rsm, mplot = TRUE, legpos = FALSE, ask = FALSE)
```

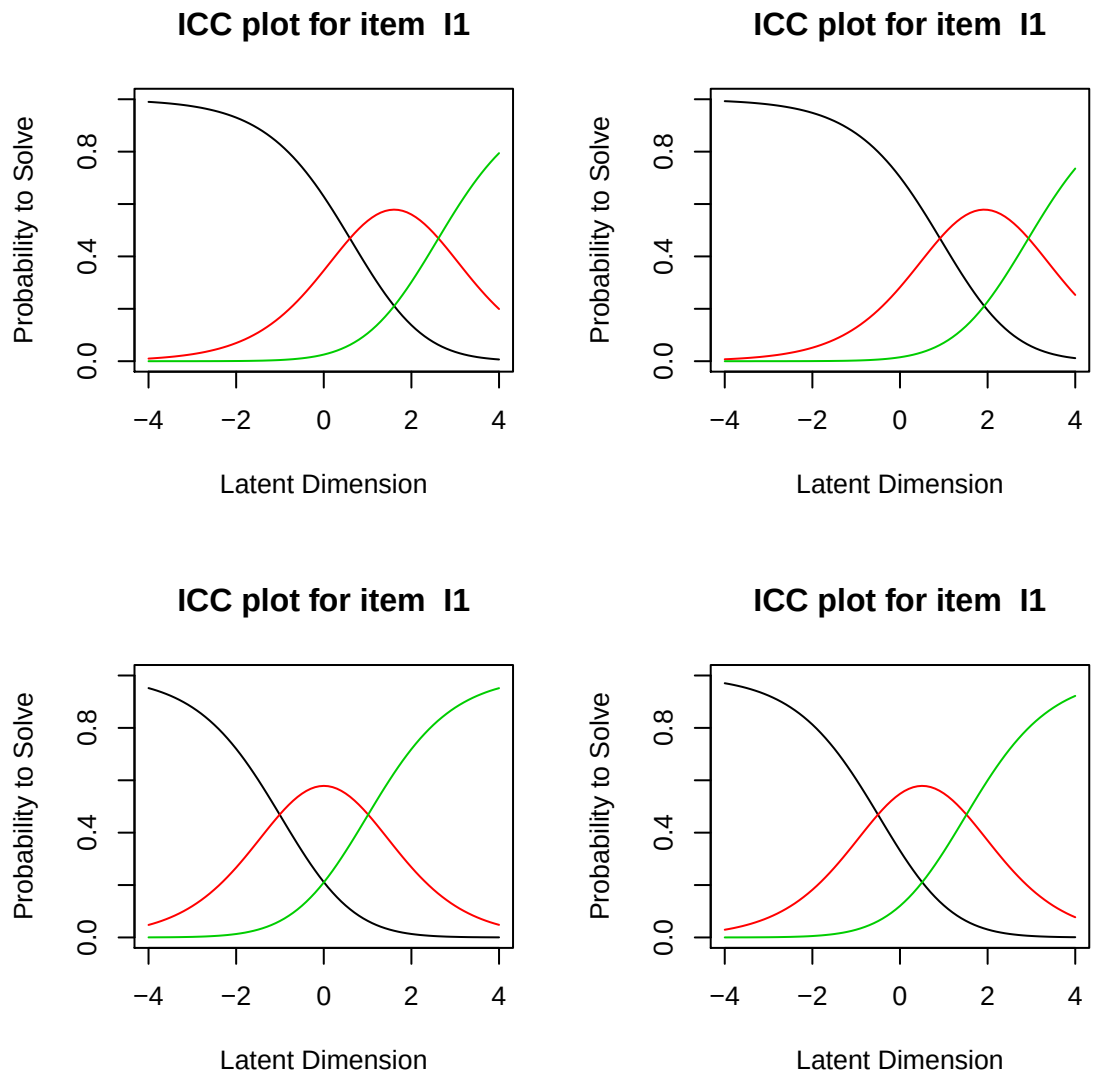


Abb. 6.4 ICC plot for an RSM.

The RSM restricts the threshold distances to be the same across all items. This strong assumption can be relaxed using a PCM. The results are represented in a person-item map (see Figure 6.5).

```
> res.pcm <- PCM(pcmdat2)
> plotPImap(res.pcm, sorted = TRUE)
```

After estimating the person parameters we can check the item-fit statistics.

```
> pres.pcm <- person.parameter(res.pcm)
> itemfit(pres.pcm)
```

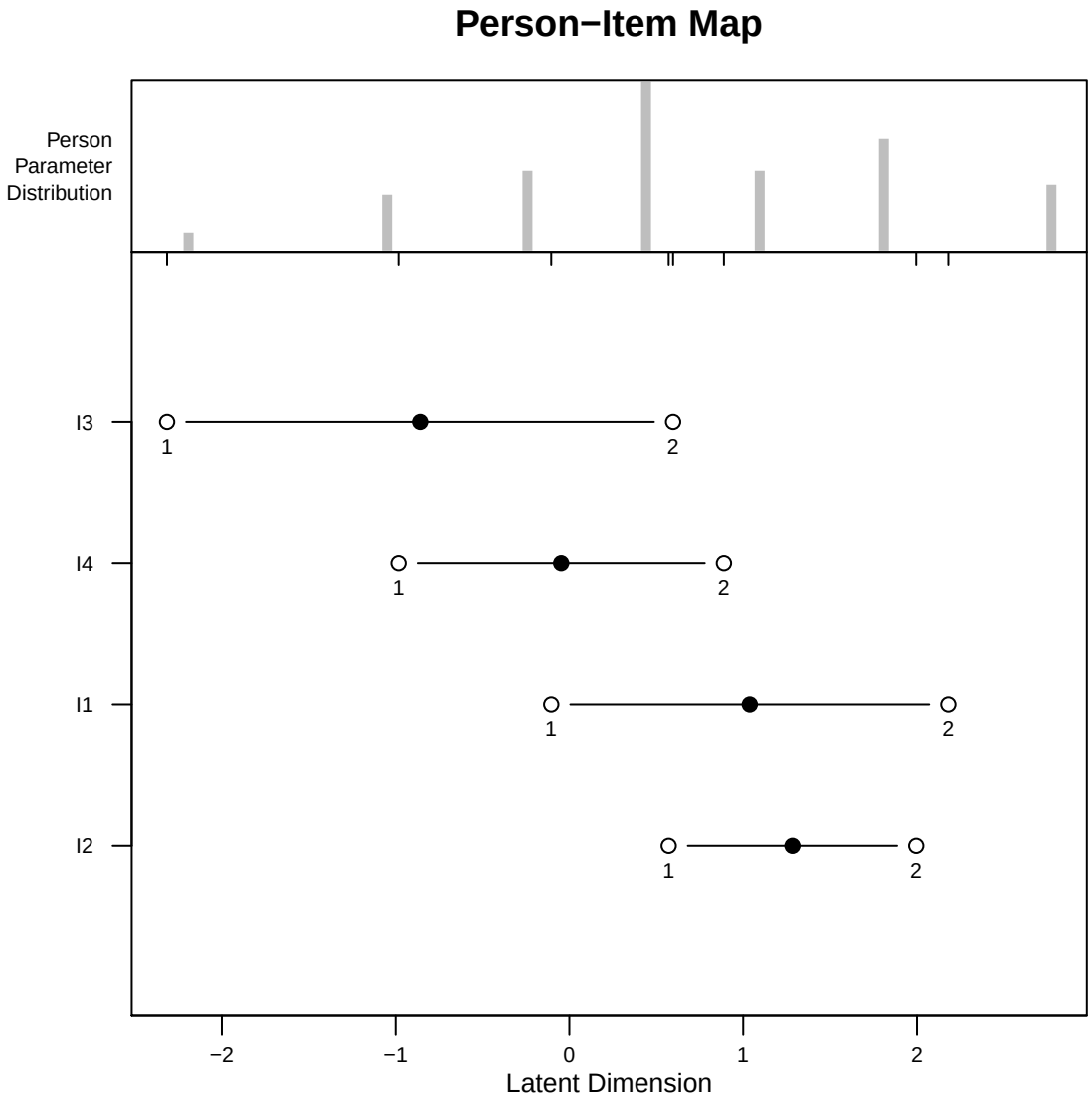


Abb. 6.5 Person-Item map for a PCM.

```
Itemfit Statistics:
     Chisq  df p-value Outfit MSQ Infit MSQ Outfit t Infit t
I1 225.617 255   0.907      0.881     0.885    -2.31   -2.29
I2 215.948 255   0.964      0.844     0.903    -2.69   -1.89
I3 179.811 255   1.000      0.702     0.713    -5.20   -5.73
I4 214.473 255   0.969      0.838     0.809    -2.80   -3.76
```

A likelihood ratio test comparing the RSM and the PCM indicates that the PCM provides a better fit.

```
> lr <- 2 * (res.pcm$loglik - res.rsm$loglik)
> df <- res.pcm$npar - res.rsm$npar
> pvalue <- 1 - pchisq(lr, df)
> cat("LR statistic: ", lr, "  df =", df, "  p =", pvalue, "\n")

LR statistic:  11.69992   df = 3   p = 0.00848509
```

6.5.5 An LPCM for repeated measurements in different groups

The most complex example refers to an LPCM with two measurement points. In addition, the hypothesis is of interest whether the treatment has an effect. The corresponding contrast is the last column in **W** below.

First, the data matrix **X** is specified. We assume an artificial test consisting of $k = 3$ items which was presented twice to the subjects. The first 3 columns in **X** correspond to the first test occasion, whereas the last 3 to the second occasion. Generally, the first k columns correspond to the first test occasion, the next k columns for the second, etc. In total, there are $n = 20$ subjects. Among these, the first 10 persons belong to the first group (e.g., control), and the next 10 persons to the second group (e.g., treatment). This is specified by a group vector:

```
> data(lpcmdat)
> grouplpcm <- rep(1:2, each = 10)
```

Again, W is generated automatically. In general, for such designs the generation of W consists first of the item contrasts, followed by the time contrasts and finally by the group main effects except for the first measurement point (due to identifiability issues, as already described).

```
> reslpcm <- LPCM(lpcmdat, mpoints = 2, groupvec = grouplpcm, sum0 = FALSE)
> model.matrix(reslpcm)
```

	eta 1	eta 2	eta 3	eta 4	eta 5	eta 6	eta 7	eta 8	eta 9	eta 10
I1.c1 t1 g1	0	0	0	0	0	0	0	0	0	0
I1.c2 t1 g1	1	0	0	0	0	0	0	0	0	0
I1.c3 t1 g1	0	1	0	0	0	0	0	0	0	0
I2.c1 t1 g1	0	0	1	0	0	0	0	0	0	0
I2.c2 t1 g1	0	0	0	1	0	0	0	0	0	0
I2.c3 t1 g1	0	0	0	0	1	0	0	0	0	0
I3.c1 t1 g1	0	0	0	0	0	1	0	0	0	0
I3.c2 t1 g1	0	0	0	0	0	0	1	0	0	0
I3.c3 t1 g1	0	0	0	0	0	0	0	1	0	0
I1.c1 t1 g2	0	0	0	0	0	0	0	0	0	0
I1.c2 t1 g2	1	0	0	0	0	0	0	0	0	0
I1.c3 t1 g2	0	1	0	0	0	0	0	0	0	0
I2.c1 t1 g2	0	0	1	0	0	0	0	0	0	0
I2.c2 t1 g2	0	0	0	1	0	0	0	0	0	0
I2.c3 t1 g2	0	0	0	0	1	0	0	0	0	0
I3.c1 t1 g2	0	0	0	0	0	1	0	0	0	0
I3.c2 t1 g2	0	0	0	0	0	0	1	0	0	0
I3.c3 t1 g2	0	0	0	0	0	0	0	1	0	0
I1.c1 t2 g1	0	0	0	0	0	0	0	0	1	0
I1.c2 t2 g1	1	0	0	0	0	0	0	0	2	0
I1.c3 t2 g1	0	1	0	0	0	0	0	0	3	0
I2.c1 t2 g1	0	0	1	0	0	0	0	0	1	0
I2.c2 t2 g1	0	0	0	1	0	0	0	0	2	0
I2.c3 t2 g1	0	0	0	0	1	0	0	0	3	0
I3.c1 t2 g1	0	0	0	0	0	1	0	0	1	0
I3.c2 t2 g1	0	0	0	0	0	0	1	0	2	0
I3.c3 t2 g1	0	0	0	0	0	0	0	1	3	0
I1.c1 t2 g2	0	0	0	0	0	0	0	0	1	1
I1.c2 t2 g2	1	0	0	0	0	0	0	0	2	2
I1.c3 t2 g2	0	1	0	0	0	0	0	0	3	3
I2.c1 t2 g2	0	0	1	0	0	0	0	0	1	1
I2.c2 t2 g2	0	0	0	1	0	0	0	0	2	2
I2.c3 t2 g2	0	0	0	0	1	0	0	0	3	3
I3.c1 t2 g2	0	0	0	0	0	1	0	0	1	1
I3.c2 t2 g2	0	0	0	0	0	0	1	0	2	2
I3.c3 t2 g2	0	0	0	0	0	0	0	1	3	3

The parameter estimates are the following:

```
Results of LPCM estimation:

Call:  LPCM(X = lpcmdat, mpoints = 2, groupvec = grouplpcm, sum0 = FALSE)

Conditional log-likelihood: -103.7225
Number of iterations: 37
Number of parameters: 10

Basic Parameters eta:
             eta 1     eta 2      eta 3      eta 4     eta 5      eta 6
Estimate -0.461590 -1.609589 -0.5713666 -0.8388421 -1.739492 -0.7232787
Std.Err   0.734666  1.194346  0.6232672  0.9854781  1.438195  0.6534237
              eta 7     eta 8      eta 9    eta 10
Estimate -0.7096128 -1.209864 -0.2014868 1.0940434
Std.Err   0.9862383  1.414828  0.2608239 0.3870401
```

Testing whether the η-parameters equal 0 is mostly not of relevance for those parameters referring to the items (in this example $\eta_1, ..., \eta_8$). But for the remaining contrasts, $H_0 : \eta_9 = 0$ (implying no general time effect) can not be rejected ($p = .44$), whereas hypothesis $H_0 : \eta_{10} = 0$ has to be rejected ($p = .004$) when applying a z-test. This suggests that there is a significant treatment effect over the measurement points. If a user wants to perform additional tests such as a Wald test for the equivalence of two η-parameters, the `vcov` method can be applied to get the variance-covariance matrix.

6.6 Additional topics

This section will be extended successively with new developments and components which do not directly relate to the modeling core of `eRm` but may prove to be useful add-ons.

6.6.1 The eRm simulation module

A recent `eRm` development is the implementation of a simulation module to generate 0-1 matrices for different Rasch scenarios. In this article we give a brief overview about the functionality and for more detailed descriptions (within the context of model testing) it is referred to Mair (2006) and Suárez-Falcón und Glas (2003).

For each scenario the user has the option either to assign θ and β as vectors to the simulation function (e.g. by drawing parameters from a uniform distribution) or to let the function draw the parameters from a $N(0, 1)$ distribution. The first scenario is the simulation of Rasch homogenous data by means of the function `sim.rasch()`. The parameter values are plugged into equation 6.3 and it results the matrix $\mathbf{P}$ of model probabilites which is of dimension $n \times k$. An element p_{vi} indicates the probability that subject v solves item i. In a second step the matrix $\mathbf{P}$ has to be transformed into the 0-1 data matrix $\mathbf{X}$. The recommended way to achieve this is to draw another random number $p^\star_{vi}$ from a uniform distribution in $[0; 1]$ and perform the transformation according to the following rule:

$$x_{vi} = \begin{cases} 1 \text{ if } p^\star_{vi} \leq p_{vi} \\ 0 \text{ if } p^\star_{vi} > p_{vi} \end{cases}$$

Alternatively, the user can specify a fixed cutpoint $p^\star := p^\star_{vi}$ (e.g. $p^\star = 0.5$) and make the decision according to the same rule. This option is provided by means of the `cutpoint` argument. Caution is advised when using this deterministic option since this leads likely to ill-conditioned data matrices.

The second scenario in this module regards the violation of the parallel ICC assumption which leads to the two-parameter logistic model (2-PL) proposed by Birnbaum (1968):

$$P(X_{vi} = 1) = \frac{\exp(\alpha_i(\theta_v - \beta_i))}{1 + \exp(\alpha_i(\theta_v - \beta_i))}. \tag{6.25}$$

The parameter α_i denotes the item discrimination which for the Rasch model is 1 across all items. Thus, each item score gets a weight and the raw scores are not sufficient anymore. The function for simulating 2-PL data is `sim.2pl()` and if $\boldsymbol{\alpha}$ is not specified by the user by means of the argument `discrim`, the discrimination parameters are drawn from a log-normal distribution. The reasons for using this particular kind of distribution are the following: In the case of $\alpha_i = 1$ the ICC are Rasch consistent. Concerning the violations, it should be possible to achieve deviations in both directions (for $\alpha_i > 0$). If $\alpha_i > 0$ the ICC is steeper than in the Rasch case and, consequently, if $\alpha_i < 1$ the ICC is flatter. This bidirectional deviation around 1 is warranted by the lognormal distribution $LN(\mu, \sigma^2)$ with $\mu = 0$. Since it is a logarithmic distribution, α_i cannot be negative. The degrees of model violation can be steered by mea ns of the dispersion parameter σ^2. A value of $\sigma^2 = .50$ already denotes a strong violation. The lower σ^2, the closer the values lie around 1. In this case the α_i are close to the Rasch slopes.

Using the function `sim.xdim()` the unidimensionality assumptions is violated. This function allows for the simulation of multidimensional Rasch models as for instance given Glas (1992) and Adams et al. (1997a). Multidimensionality implies that one single item measures more than one latent construct. Let us denote the number of these latent traits by D. Consequently, each person has a vector of ability parameters $\boldsymbol{\theta}_v$ of length D. These vectors are drawn from a multivariate normal distribution with mean $\boldsymbol{\mu} = 0$ and VC-matrix Σ of dimension $D \times D$. This matrix has to be specified by the user with the argument `Sigma`. In order to achieve strong model violations, very low correlations such as .01 should be provided. To specify to which extend item i is measuring each of the D dimensions, a corresponding vector of weights $\mathbf{z}_i$ of length D is defined. If the resulting $k \times D$ matrix $\mathbf{Z}$ is not provided by the user, `sim.xdim()` generates $\mathbf{Z}$ such that each $\mathbf{z}_i$ contains only nonzero element which indicates the assigned dimension. This corresponds to the *between-item multidimensional model* (Adams et al., 1997a). However, in any case the person part of the model is $\mathbf{z}_i^T \boldsymbol{\theta}_v$ which replaces θ_v in Equation 6.3.

Finally, locally dependent item responses can be produced by means of the function `sim.locdep()`. Local dependence implies the introduction of pair-wise item correlations δ_{ij}. If these correlations are constant across items, the argument `it.cor` can be a single value δ. A value $\delta = 0$ corresponds to the Rasch model whereas $\delta = 1$ leads to the strongest violation. Alternatively, for different pair-wise item correlations, the user can specify a VC-matrix Δ of dimension $k \times k$. The formal representation of the corresponding IRT model is

$$P(X_{vi} = 1 | X_{vj} = x_{vj}) = \frac{\exp(\theta_v - \beta_i + x_{vj}\delta_{ij})}{1 + \exp(\theta_v - \beta_i + x_{vj}\delta_{ij})}. \tag{6.26}$$

This model was proposed by Jannarone (1986) and is suited to model locally dependent item responses.

6.7 Discussion and outlook

Here we give a brief outline of future `eRm` developments. The CML estimation approach, in combination with the EM-algorithm, can also be used to estimate *mixed Rasch models* (MIRA). The basic idea behind such models is that the extended Rasch model holds within subpopulations of individuals, but with different parameter values for each subgroup. Corresponding elaborations are given in Rost und von Davier (1995).

In Rasch models the item discrimination parameter α_i is always fixed to 1 and thus it does not appear in the basic equation. Allowing for different discrimination parameters across items leads to the two-parameter logistic model as given in Equation 6.25. In this model the raw scores are not sufficient statistics anymore and hence CML can not be applied. 2-PL models can be estimated by means of the `ltm` package (Rizopoulos, 2006). However, Verhelst und Glas (1995) formulated the one parameter logistic model (OPLM) where the α_i do not vary across the items but are unequal to one. The basic strategy to estimate OPLM is a three-step approach: First, the item parameters of the Rasch model are computed. Then, discrimination parameters are computed under certain restrictions.

Finally, using these discrimination weights, the item parameters for the OPLM are estimated using CML. This is a more flexible version of the Rasch model in terms of different slopes.

To conclude, the eRm package is a tool to estimate extended Rasch models for unidimensional traits. The generalizations towards different numbers of item categories, linear extensions to allow for introducing item covariates and/or trend and optionally group contrasts are important issues when examining item behavior and person performances in tests. This improves the feasibility of IRT models with respect to a wide variety of application areas.

Kapitel 7
ACER ConQuest – Generalised Item Response Modelling Software

Bernadett Pauer, Julia Anette von Poswik und Thomas Rusch

Zusammenfassung In folgendem Kapitel wird das Programm ACER ConQuest vorgestellt. Mit ConQuest können verschiedene Item Response Modelle und latente Regressionsmodelle nachvollzogen werden. Im Gegensatz zu anderen Programmen mit Fokus auf Modellierungen im Bereich der Item Response Theorie, ist ACER ConQuest zusätzlich in der Lage Multidimensionale Item Response Modelle nachzubilden. Das Kapitel widmet sich zuerst den generellen Anwendungsbereichen und den Eigenschaften von ACER ConQuest um dann drei beispielhafte Modellierungen vorzustellen. In den Anwendungsbeispielen wird jeweils kurz auf die Forschungsfragen und Rahmenbedingungen der Untersuchungen eingegangen. Die Befehle zur Modellierung der Item Response Theorien werden beschrieben und die Ergebnisse der Berechnungen mit ACER ConQuest vorgestellt. Als Beispiele wurden ausgewählt: Die Modellierung von dichotomen Multiple Choice Tests (Beispiel 1), von Beurteilungseffekten (Beispiel 2) und von mehrdimensionalen latenten Regressionen (Beispiel 3).

7.1 ACER ConQuest - Eine Einführung

> ConQuest brings together in a single program a wide variety of item response models (...) and provides an integration of item response and regression analysis. (Wu et al., 2007)

ACER ConQuest, ein Programm zur Modellierung verschiedener Ansätze der Item Response Theory (IRT), liegt seit dem Jahr 2007 in der Version 2.0 vor. Gegenüber der Version 1.0 aus dem Jahr 1998 zeichnet sich die neue Auflage von ACER ConQuest laut Autoren durch eine höhere Benutzerfreundlichkeit und Effizienz sowie eine erweiterte Funktionalität aus. Die Software wurde vom Australian Council for Educational Research Ltd (ACER) im Rahmen der PISA (Programme for International Student Assessment) Studien entwickelt. Die verantwortlichen Autoren der Software sowie des dazugehörigen Manuals sind Margeret L. Wu, Raymond J. Adams, Mark R. Wilson und Samuel A. Haldane. Das vorliegende Kapitel bezieht sich in weiten Strecken auf das Manual dieser Autoren zur Version 2.0 von ConQuest (Wu et al., 2007).

7.1.1 Modellierungen in ConQuest

ConQuest eignet sich zur Modellierung verschiedener Item Response und latenten Regressionsmodellen. Mithilfe folgender Modelle können psychometrische Daten mit ConQuest analysiert werden:

- Raschs einfaches logistisches Modell für dichotome Items (Rasch, 1980)
- Das Rating Skalen Modell für Items mit einem gemeinsamen, mehrkategoriellen Antwortformat (Andrich, 1978b)

WU Wien, Institut für Change Management und Management Development • bernadett.pauer@googlemail.com
WU Wien, Institut für Personalmanagement • juliavposwik@gmx.net
WU Wien, Institute for Statistics and Mathematics • thomas.rusch@wu.ac.at

- Das Partial Credit Model für mehrere spezifische Anwortkategorien pro Item (Masters, 1982)
- Das Ordered Partition Model, bei welchem mehrere Antwortkategorien denselben Skalenwert erhalten können (Wilson, 1992)
- Das Linear Logistische Test Model, welches die Itemparameter als lineare Kombination von elementaren Parametern niedrigerer Ordnung auffasst (Fischer, 1983)
- Multifaceted Models, welche zusätzlich zum Personen- und Itemparameter weitere Parameter zulassen, wie z.B. einen *Rater*-Parameter, um die Subjektivität in der Beurteilungen von Leistungen zu berücksichtigen (Linacre, 1994a)
- Generalisierte eindimensionale Modelle, welche Kombinationen von Antworten mit zwei und mehreren Facetten zulassen
- Multidimensionale Item Response Modelle, die Verhaltensantworten auf bis zu zehn latente Dimensionen zurück führen (Wang, 1995) (Adams et al., 1997b)
- Latente Regressionsmodelle zur direkten Schätzung von Regressionen (Adams et al., 1997b).

ConQuest schätzt die Parameter der genannten Modell durch die Marginal Maximum Likelihood Methode (MML) und antizipiert hierbei für die Personenparameter eine Normalverteilung. ConQuest stellt zudem basierend auf den Item Response Modellierungen Itemanalysen bereit und erstellt neben den Parameterschätzungen auch Schätzfehler und Fit-Indizes. Auch Itemstatistiken gemäß der klassischen Testtheorie können in ConQuest berechnet werden. Neben diesen grundsätzlichen Informationen über die Itemeigenschaften, kann mit ConQuest auch die Gleichwertigkeit der Itemparameter in unterschiedlichen Gruppen untersucht werden (Item-Differentiale). Ferner lässt ConQuest die Berechnung der Konsistenz von Beurteilungen in Beobachtungskontexten zu (*Rater*-Effekte). Während die genannten Anwendungen auch in anderen Programmen bereit gestellt werden, bietet ConQuest zusätzlich auch die Möglichkeit Multidimensionale Item Response Modelle zu prüfen und die Korrelationen zwischen den latenten Variablen zu schätzen (Wu et al., 2007). Während die meisten IRT Modelle eine Eindimensionalität der zugrunde liegenden Dimension voraussetzen, lassen Multidimensionale Item Response Modelle mehrere latente Dimensionen als Erklärungsbasis für ein gezeigtes Verhalten zu. Die Relevanz dieser Anwendung gründet sich einerseits darauf, dass viele standardisierte, psychometrische Tests unterschiedliche und unkorrelierte Persönlichkeitseigenschaften erfassen sollen und andererseits die Tendenz der diagnostischen Praxis dahin geht, ein gezeigtes Verhalten aus unterschiedlichen Perspektiven untersuchen zu wollen. Das Multidimensional Random Coefficients Multinominal Logit Model (MRCMLM) nimmt daher an, dass ein Set von D Persönlichkeitseigenschaften die Basis für das Verhalten einer Person darstellt. Diese D latenten Eigenschaften spannen einen D-dimensionalen, latenten Raum auf, in welchem die Lage der Person über einen mehrdimensionalen Personenparameter und -vektor definiert ist $\theta = \theta_1 + \theta_2 + \theta_3 + \ldots + \theta_D$ Adams et al. (1997a).

7.2 Modell und Schätzung

7.2.1 Das multidimensionale multinominale Logit-Modell mit zufallsbedingten Koeffizienten

Bei der multidimensionalen Form handelt es sich um eine Erweiterung des unidimensionalen Modells. Dabei wird es angenommen, dass ein Set von D latenten Merkmalen den Antworten der Individuen zugrunde liegt. Die D latenten Merkmale definieren eine D-dimensionale Fläche, wobei die Positionen der einzelnen Individuen mit dem Vektor $\theta = \theta_1 + \theta_2 + \ldots + \theta_D$ repräsentiert sind. Die Wahrscheinlichkeit einer Antwort in der Kategorie k auf Items i wird folgenderweise modelliert:

$$Pr(\mathbf{X}_{ij} = 1; \mathbf{A}, \mathbf{B}, \xi|\theta) = \frac{\exp(\mathbf{b}'_{ij}\theta + \mathbf{a}'_{ij}\xi)}{\sum_{k=1}^{K_i} \exp(\mathbf{b}'_{ik}\theta + \mathbf{a}'_{ik}\xi)} \tag{7.1}$$

Als Antwortvektor erhält man dementsprechend:

$$f(\mathbf{x};\xi|\theta) = \Psi(\theta,\xi)\exp[\mathbf{x}'(\mathbf{B}\theta + \mathbf{A}\xi)], \tag{7.2}$$

mit

$$\Psi(\theta,\xi) = \left\{\sum_{z\in\Omega}\exp[\mathbf{z}'(\mathbf{B}\theta + \mathbf{A}\xi)]\right\}^{-1} \tag{7.3}$$

Zwischen dem unidimensionalen und multidimansionalen Modell besteht der Unterschied darin, dass im unidimensionalen Modell die Fähigkeitsparameter ein skalares θ ist, während im multidimensionalen Modell das θ einen $D \times 1$-Spaltenvektor darstellt. Der gleiche Unterschied besteht auch bezüglich der Ergebnisfunktion der Antwort k auf Item i.

Zur Identifizierung des Antwortvektors müssen die Matrizen A und B bestimmte Bedingungen erfüllen, wozu die Bedingungen von Volodin und Adams (1995) verwendet werden können.

7.2.2 Das Populationsmodell

Das Item Response Modell ist ein konditionales Modell, welches beschreibt, wie Item Responses von einer latenten Variablen θ abhängen. Die vollständige Definition des Modells basiert auf der Dichtefunktion

$$f_\theta(\theta;\alpha) \tag{7.4}$$

für die latente Variable θ . α symbolisiert ein Set von Parametern, welche die Verteilung der Variable θ charakterisieren. Im Regelfall wird eine Normalverteilung der abhängigen Variablen angenommen. Unter der Annahme einer normalverteilten abhängigen Variablen θ , kann die Schätzung des Populationsmodells mit den korrespondierenden Regressionsparametern bestimmter Personvariablen β , der Populationsvarianz σ^2 und dem Vektor des Itemparameters ξ erfolgen.

$$f_\theta(\theta_n;\mathbf{W}_n,\gamma,\Sigma) = (2\pi)^{-\frac{d}{2}}|\Sigma|^{-\frac{1}{2}}\exp\left[-\frac{1}{2}(\theta_n - \gamma\mathbf{W}_n)'\Sigma^{-1}(\theta_n - \gamma\mathbf{W}_n)\right] \tag{7.5}$$

Das Populationsmodell ist multivariat und beschreibt die Abhängigkeit der Variablen θ von mehreren Personvariablen. γ entspricht einer Matrix von Regressionskoeffizienten dieser Personvariablen, Σ einer Varianz-Kovarianzmatrix und W_n einem Vektor definierter Personvariablen. Wenn die beschriebene Formel als Populationsmodell verwendet wird, können die Parameter γ, Σ und ξ geschätzt werden.

7.2.3 Schätzung

Kombinieren des bedingten Item-Response-Modells (7.1) und des Populationsmodells (7.5) führt zum unbedingten Item-Response-Modell

$$f_x(\mathbf{x};\xi,\gamma,\Sigma) = \int_\theta f_x(\mathbf{x};\xi|\theta)f_\theta(\theta;\gamma,\Sigma)d\theta \tag{7.6}$$

und somit ist die Likelihood gegeben als

$$\Lambda = \prod_{n=1}^{N} f_x(\mathbf{x};\xi,\gamma,\Sigma) \tag{7.7}$$

wobei N die Gesamtzahl der Subjekte bezeichnet.

Partielles Differenzieren nach allen Unbekannten, ξ, γ, Σ ergibt die Schätzgleichungen

$$A' \sum_{n=1}^{N} \left[\mathbf{x}_n - \int_{\theta_n} E_z(\mathbf{z}|\theta_n) h_\theta(\theta_n; \mathbf{Y}_n, \xi, \gamma, \Sigma|\mathbf{x}_n) d\theta_n \right] = 0, \tag{7.8}$$

$$\hat{\gamma} = \left(\sum_{n=1}^{N} \bar{\theta}_n \mathbf{W}_n' \right) \left(\sum_{n=1}^{N} \mathbf{W}_n \mathbf{W}_n' \right)^{-1}, \tag{7.9}$$

sowie

$$\hat{\Sigma} = \frac{1}{N} \sum_{n=1}^{N} \int_{\theta_n} (\theta_n - \gamma \mathbf{W}_n)(\theta_n - \gamma \mathbf{W}_n)' h_\theta(\theta_n; \mathbf{Y}_n, \xi, \gamma, \Sigma|\mathbf{x}_n) d\theta_n. \tag{7.10}$$

Hierbei ist

$$h_\theta(\theta_n; \mathbf{Y}_n, \xi, \gamma, \Sigma|\mathbf{x}_n) = \frac{f_x(\mathbf{x}_n; \xi|\theta_n) f_\theta(\theta_n; \mathbf{W}_n, \gamma, \Sigma)}{f_x(\mathbf{x}_n; \mathbf{W}_n, \xi, \gamma, \Sigma)} \tag{7.11}$$

die marginale Posterior und

$$E_z(\mathbf{z}|\theta_n) = \Psi(\theta_n, \xi) \sum_{z \in \Omega} \mathbf{z} \exp[\mathbf{z}'(\mathbf{b}\theta_n + \mathbf{A}\xi)], \tag{7.12}$$

sowie

$$\bar{\theta}_n = \int_{\theta_n} \theta_n h_\theta(\theta_n; \mathbf{Y}_n, \xi, \gamma, \Sigma|\mathbf{x}_n) \tag{7.13}$$

Diese Gleichungssysteme werden in ConQuest mittels EM-Algorithmus gelöst.

7.2.4 Der Aufbau von ACER Conquest

ConQuest kann sowohl über eine eher benutzerfreundliche, graphische Oberfläche (graphical user interface, GUI) oder eine einfache, dafür aber in den administrierten Berechnungen schnellere Kommandoplattform (console interface, CMD) bedient werden. Beide Versionen laufen unter Windows 98, 2000 und XP. Generell wird in der GUI Version ein Input Window unterschieden, in welchem die Daten eingetragen und die Berechnungen veranlasst werden können, und einem Output Window, in welchem der Fortschritt und die Ergebnisse der Berechnungen angezeigt werden. Über eine Menüleiste können die einzelnen Befehle mittels drop-down-Listen aufgerufen werden. Eine genaue Beschreibung der Benutzeroberfläche und der zur Verfügung stehenden Befehle kann im Manual von ACER ConQuest nachgelesen werden (Wu et al., 2007).

7.3 Anwendungen

7.3.1 Beispiel 1: Dichotomer Multiple Choice Test

Der Multiple Choice Test stellt ein verbreitetes Verfahren zur Prüfung von kognitiven Fähigkeiten dar. Die Analyse der Eigenschaften von dichotomen Items ist die einfachste Anwendung von ConQuest. Dieses erste Beispiel zeigt, wie ConQuest verwendet werden kann, um das Raschs einfache logistische Modell an die Daten anzupassen, die mit Hilfe eines Multiple Choice Tests gesammelt wurden. Dabei generiert ConQuest auch eine Reihe von traditionellen Test-Item-Statistiken.
In diesem Beispiel werden die folgenden Dateien verwendet:

ex1.cqc Befehle
ex1.dat Daten
ex1.lab Variable Beschriftung ("label") für die Items
ex1.shw Ergebnisse der Rasch-Analyse
ex1.itn Ergebnisse der traditionellen Item-Analyse

Die verwendeten Daten stammen aus einem Multiple Choice Test mit 12 Items, der von 1 000 Studierenden ausgefüllt wurde. Die Daten wurden in der Datei ex1.dat erfasst. Die Zeilen repräsentieren die Studierenden. Die ersten 5 Spalten beinhalten das Identifikationskennzeichen der Studierenden, die nächsten 7 Spalten sind leer und in den Spalten 12 bis 23 befinden sich die Antworten der Studierenden zu den Fragen des Multiple Choice Tests. Nicht beantwortete Fragen wurden mit M erfasst. Die Abbildung 7.1 zeigt einen Ausschnitt aus der Datei ex1.dat.

Nach Starten von ConQuest öffnet sich automatisch ein Input-Fenster. Hier sollen die Befehle

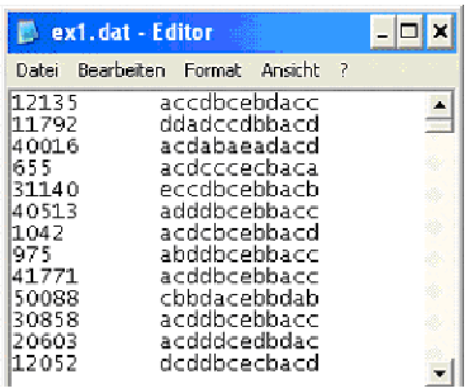


Abb. 7.1 Ausschnitt aus der Datei ex1.dat

folgenderweise eingegeben werden:

```
Datafile ex1.dat;
Format id 1-5 responses 12-23;
Labels << ex1.lab;
Key acddbcebbacc ! 1;
Model item;
Estimate;
Show >> ex1.shw;
Itanal >> ex1.itn;
Plot icc;
Plot mcc ! legend=yes
```

Datafile gibt den Namen und Speicherort der Datenfile an.

Format beschreibt das Layout der Daten in der Datei ex1.dat und gibt an, dass sich in den ersten 5 Spalten die Identifikationskennzeichen und in den Spalten 12 bis 23 die Antworten befinden. Vergisst man anzugeben, wo die Antworten zu finden sind, läuft ConQuest nicht!

Label Die Definition von Labels ist nicht erforderlich, verbessert aber die Lesbarkeit der Ergebnisse. Labels dürfen nur Testdateien sein! Die Eingabe fängt mit dem Zeichen ===> an, dann folgen ein oder mehrere Leerzeichen und der Name der Variable, die mit Hilfe der Labels beschrieben wird. Die Abbildung 7.2 zeigt einen Ausschnitt aus ex1.lab.

Key definiert die richtigen Antworten zu den einzelnen Multiple Choice Fragen. Die Zahl 1 nach dem Rufzeichen gibt an, dass die richtigen Antworten mit 1 zu bewerten sind.

Model gibt den Namen der Variable an, die die Anwortdaten identifiziert. In diesem Beispiel: Item.

Estimate ist der Befehl zur Schätzung des Item Response Modells.

Show Das Zeichen » weist darauf hin, dass die Ergebnisse in der Datei ex1.shw zusammengefasst werden sollen. Verzichtet man auf diesen Befehl, dann erscheinen die Ergebnisse direkt im Output-Fenster.

Itanal Wie bei Show werden auch hier die Ergebnisse in eine externe Datei exportiert (ex1.itn). Dieser File zeigt die Ergebnisse der tratitionellen Item-Analyse.

Plot icc erstellt eine Graphik pro Item, die die modellierten und empirischen Item-Charakteristikkurven vergleicht.

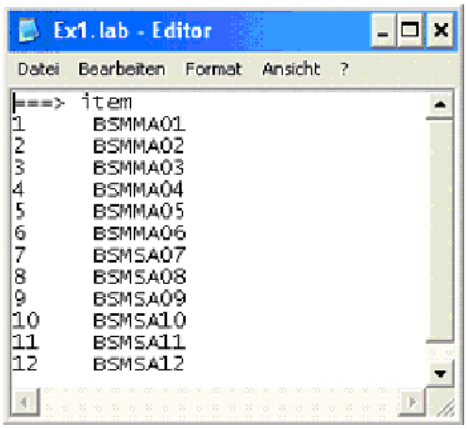


Abb. 7.2 Ausschnitt aus der Datei ex1.lab

Plot mcc erstellt eine Graphik pro Item, die einerseits die modellierten und empirischen Item-Charakteristikkurven vergleicht (richtige Antworten), andererseits das Verhalten von Distractors (falsche Antworten) darstellt.

Die Reihenfolge der Befehlszeilen ist nicht fixiert, es bestehen allerdings logische Beschränkungen. Show kann zum Beispiel nicht vor Estimate stehen und die Befehle Model, Format oder Datafile müssen noch vor Estimate definiert werden. Durch den Befehl Run wird die Modellanalyse begonnen. Im Output-Fenster erscheinen die Änderungen durch den Schätzvorgang. Die Abbildung 7.3 zeigt zum Beispiel die Änderungen nach 4 Iterationen. Nach dem Schätzvorgang werden die Da-

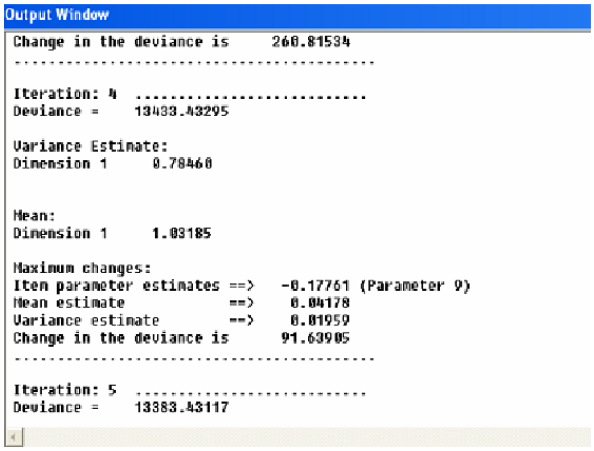


Abb. 7.3 Dichotomer Multiple Choice Test - Iterationen

teien show bzw. itanal und die Graphiken erstellt. Die Abbildung 7.4 zeigt die Datei ex1.shw. In diesem Beispiel wurden insgesamt 13 Parameter geschätzt (Mittel, Varianz und 11 Itemschwierigkeitsparameter). Es werden deswegen 11 statt 12 Itemparameter geschätzt, weil ein Parameter von ConQuest per default zu Identifikationszwecken des Modells fixiert werden muss. Der nächste Teil der Datei ex1.shw zeigt die Parameterschätzungen für jedes Test-Item mit ihren Standardfehlern und mit einigen Fit-Statistiken. Die Item-Parameterschätzungen sind Marginal Maximum Likelihood-Schätzungen, bei den Standardfehlern handelt es sich um asymptotische Schätzungen durch die Inverse der Hessenmatrix und die Fit-Statistiken stellen residual-basierte Kennzeichen dar, die mit den von Wright und Stone (1979) (für das Rasch Simple Logistic Model) und Wright und Masters (1982) (für das Partial Credit Model) entwickelten gewichteten und ungewichteten Fit-Statistiken vergleichbar sind. Für die MNSQ Fit-Statistik wird ein 95%-iges Konfidenzintervall angenommen. Wenn sich die MNSQ Fit-Statistik ausserhalb dieses Intervalls befindet, dann weist die t-Statistik einen Wert über 2,0 aus und es wird die Null-Hypothese verworfen, dass die Daten dem Modell

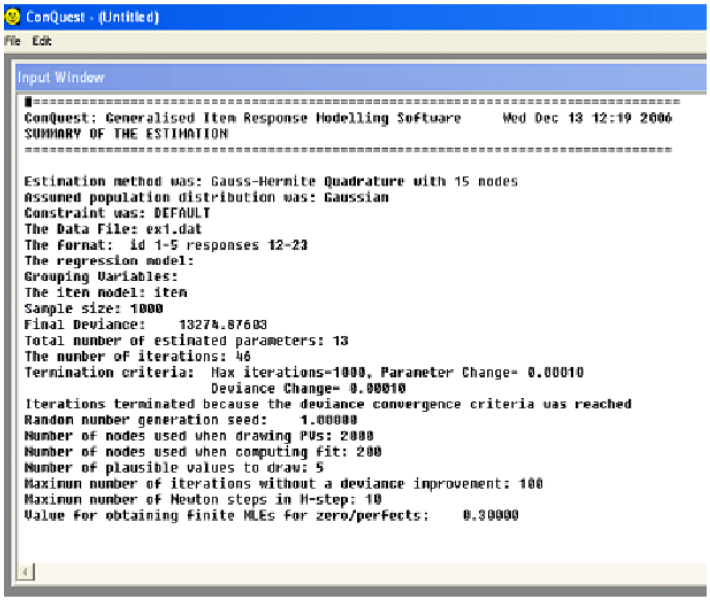

ConQuest - (Untitled)
File Edit
Input Window
ConQuest: Generalised Item Response Modelling Software Wed Dec 13 12:19 2006
SUMMARY OF THE ESTIMATION
Estimation method was: Gauss-Hermite Quadrature with 15 nodes
Assumed population distribution was: Gaussian
Constraint was: DEFAULT
The Data File: ex1.dat
The format: id 1-5 responses 12-23
The regression model:
Grouping Variables:
The item model: item
Sample size: 1000
Final Deviance: 13274.87603
Total number of estimated parameters: 13
The number of iterations: 46
Termination criteria: Max iterations=1000, Parameter Change= 0.00010
Deviance Change= 0.00010
Iterations terminated because the deviance convergence criteria was reached
Random number generation seed: 1.00000
Number of nodes used when drawing PVs: 2000
Number of nodes used when computing fit: 200
Number of plausible values to draw: 5
Maximum number of iterations without a deviance improvement: 100
Maximum number of Newton steps in M-step: 10
Value for obtaining finite MLEs for zero/perfects: 0.30000

Abb. 7.4 Dichtomer Multiple Choice Test - ex1.shw - summary of the estimation

entsprechen. Die Abbildung 7.5 zeigt die Fit-Statistiken in der Datei ex1.shw. Die Datei ex1.shw zeigt

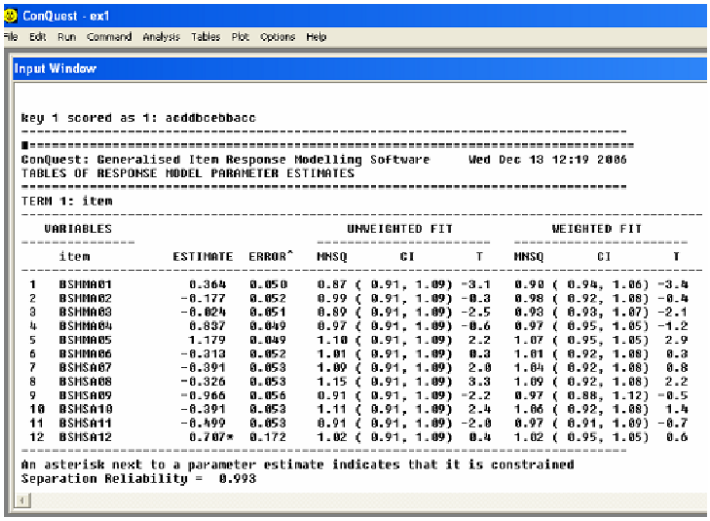


Abb. 7.5 Dichotomer Multiple Choice Test - ex1.shw - Fit statistics

auch die Item Separation Reliability, den Chi-Quadrat-Test, den Mittelwert und die Varianz. Die Item Separation Reliability gibt an, wie gut die Itemparameter voneinander trennbar sind (vgl. Wright und Stone (1979)) und hat einen Maximalwert von 1 und einen Mindestwert von 0. Die Item Separation Reliability steigt mit steigender Stichprobenanzahl. In diesem Beispiel beträgt die Item Separation Reliability 0.993 wie auch die Abbildung 7.6 zeigt. Der dritte Teil der Datei ex1.shw liefert einen Plot der Item-Schwierigkeitsparameter, wie auch in der Abbildung 7.7 dargestellt. Die Ergebnisse der traditionellen Item-Analyse werden in der Datei ex1.itn dargestellt. Diese Analyse beinhaltet die klassische Schwierigkeitsanalyse und die Diskriminierungsstatistiken bzw. die punkt-biseriale Korrelation. Die punkt-biseriale Korrelation gibt die Korrelation zwischen der Gesamtpunktezahl der Studierenden und einem Indikator an, der den Wert 1 annimmt, wenn die Frage richtig beantwortet wurde und 0, wenn nicht. Die Abbildung 7.8 zeigt einen Ausschnitt aus der Datei ex1.itn. Am Ende der Datei ex1.itn werden auch die zusammenfassenden Ergebnisse dargestellt, wie auch in der Abbildung 7.9 ersichtlich ist. Die Abbildungen 7.10 und 7.11 zeigen je eine Graphik nach plot icc und plot

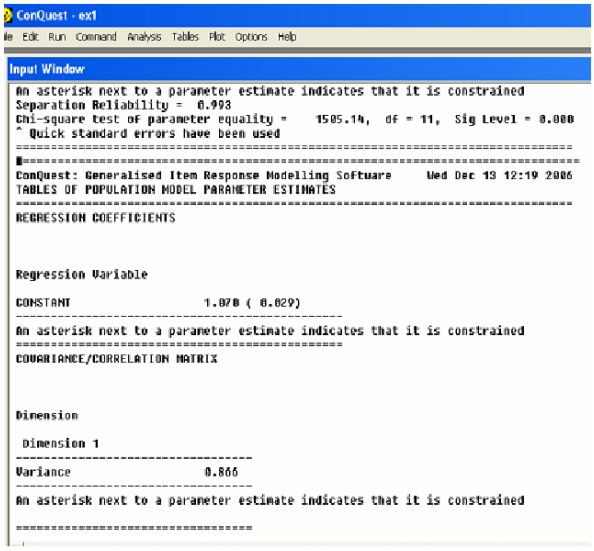

ConQuest - ex1
File Edit Run Command Analysis Tables Plot Options Help
Input Window
An asterisk next to a parameter estimate indicates that it is constrained
Separation Reliability = 0.993
Chi-square test of parameter equality = 1505.14, df = 11, Sig Level = 0.000
^ Quick standard errors have been used
ConQuest: Generalised Item Response Modelling Software Wed Dec 13 12:19 2006
TABLES OF POPULATION MODEL PARAMETER ESTIMATES
REGRESSION COEFFICIENTS
Regression Variable
CONSTANT 1.070 (0.029)
An asterisk next to a parameter estimate indicates that it is constrained
COVARIANCE/CORRELATION MATRIX
Dimension
Dimension 1
Variance 0.866
An asterisk next to a parameter estimate indicates that it is constrained

Abb. 7.6 Dichotomer Multiple Choice Test - ex1.shw - Item Separation Reliability, Chi-squared Test, Mean, Variance

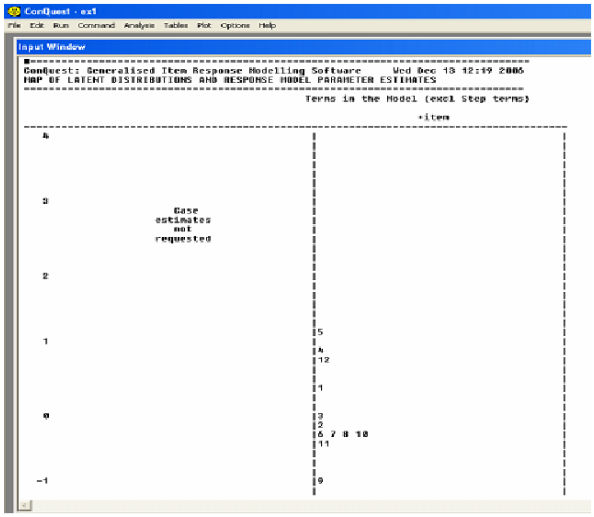

ConQuest - ex1
Input Window
ConQuest: Generalised Item Response Modelling Software Wed Dec 13 12:19 2006
MAP OF LATENT DISTRIBUTIONS AND RESPONSE MODEL PARAMETER ESTIMATES
Terms in the Model (excl Step terms)
+item
Base estimates not requested

Abb. 7.7 Dichotomer Multiple Choice Test -ex1.shw - Item and latent distribution map for simple logistic model

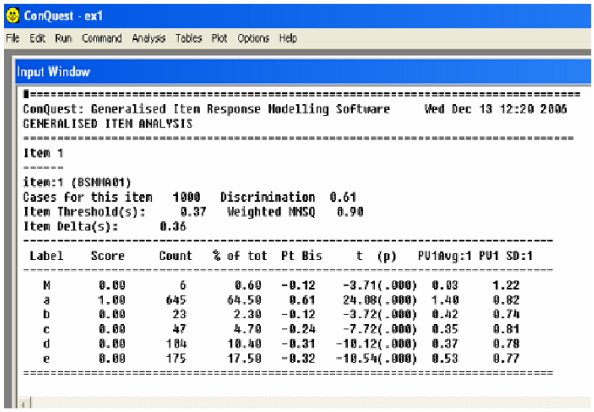

ConQuest - ex1
File Edit Run Command Analysis Tables Plot Options Help
Input Window
ConQuest: Generalised Item Response Modelling Software Wed Dec 13 12:20 2006
GENERALISED ITEM ANALYSIS
Item 1
item:1 (BSMMA01)
Cases for this item 1000 Discrimination 0.61
Item Threshold(s): 0.37 Weighted MNSQ 0.90
Item Delta(s): 0.36
Label Score Count % of tot Pt Bis t (p) PV1Avg:1 PV1 SD:1
M 0.00 6 0.60 -0.12 -3.71(.000) 0.88 1.22
a 1.00 645 64.50 0.61 24.08(.000) 1.40 0.82
b 0.00 23 2.30 -0.12 -3.72(.000) 0.42 0.74
c 0.00 47 4.70 -0.24 -7.72(.000) 0.35 0.81
d 0.00 184 18.40 -0.31 -10.12(.000) 0.37 0.78
e 0.00 175 17.50 -0.32 -10.54(.000) 0.53 0.77

Abb. 7.8 Dichtomer Multiple Choice Test - ex1.itn - Difficulty, discrimination, point-biserial statistics

mcc. Diese Graphiken werden pro Item erzeugt und vergleichen die modellierten und empirischen Itemcharakteristiken bzw. stellen das Verhalten von Distraktoren dar. Eine genauere Beschreibung des Beispiels kann im Manual von ACER ConQuest nachgelesen werden (Wu et al., 2007).

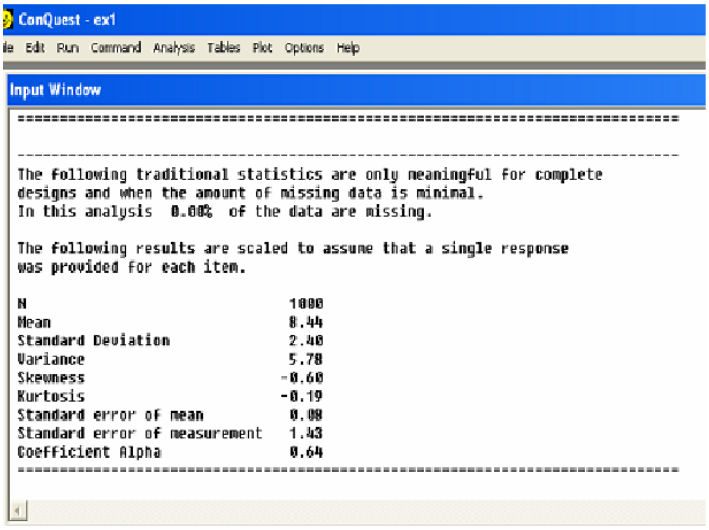

ConQuest - ex1
File Edit Run Command Analysis Tables Plot Options Help
Input Window
The following traditional statistics are only meaningful for complete designs and when the amount of missing data is minimal.
In this analysis 0.00% of the data are missing.
The following results are scaled to assume that a single response was provided for each item.
N 1000
Mean 8.44
Standard Deviation 2.40
Variance 5.78
Skewness -0.60
Kurtosis -0.19
Standard error of mean 0.08
Standard error of measurement 1.43
Coefficient Alpha 0.64

Abb. 7.9 Dichotomer Multiple Choice Test - ex1.itn - Summary results

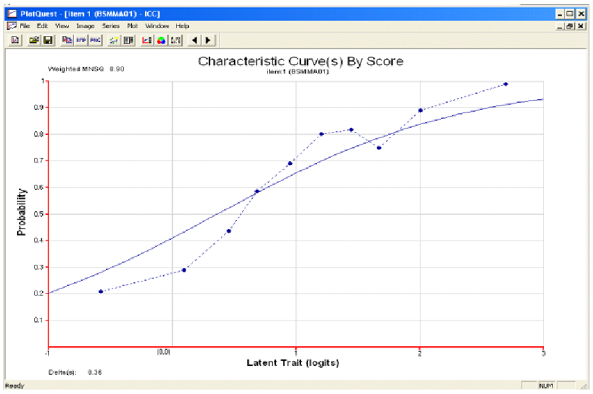


Abb. 7.10 Dichotomer Multiple Choice Test - plot icc

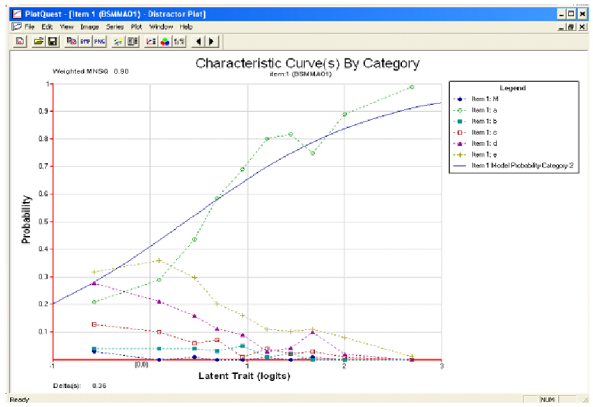


Abb. 7.11 Dichotomer Multiple Choice Test - plot mcc

7.3.2 Beispiel 2: Die Analyse von Beurteilungseffekten

Beurteilungen von Verhaltensantworten finden in einem Messkontext statt, der durch drei Facetten oder Einflüsse geprägt wird:

1. Die Leistung einer Person ist das Messobjekt und lässt sich mithilfe eines Personenparameters bestimmen.

2. Die Messung erfolgt zum einen über eine Aufgabe oder ein Item, welches über eine Schwierigkeit verfügt und daher ber einen Itemparameter verf ügt lässt.
3. Zum anderen erfolgt die Messung aber auch in Form einer Beurteilung durch einen *Rater*, welcher in seinem Urteil eher streng oder milde sein kann. Die Urteile unterschiedlicher Rater über die gleiche Leistung können deswegen konsistenter oder weniger konsistent ausfallen. Linacre (1994a)

Im Manual zu ACER ConQuest wird ein experimentelles Setting beschrieben, durch welches gezeigt wird, wie ACER ConQuest zur Untersuchung von Beurteilungsgeffekten beitragen kann: Untersucht wurden die Leistungen von rund 8300 Schülern in einem Aufsatz, wobei jeder Aufsatz von jeweils zwei Beurteilern eingeschätzt werden sollte. Insgesamt gab es 16 *Rater*, welche jeden Aufsatz dahingehend beurteilten, wie dessen Gesamteindruck war (Overall Performance, OP) und wie der Text im Detail gestaltet war (Textual Features, TF). Beide Merkmale, OP und TF wurden auf einer 6-stufigen Skala von G bis L beurteilt. Die Daten liegen den Programmdateien als Rohtext in einem .dat-Format bei und werden von ConQuest über einen Befehl eingelesen. Um die Daten korrekt einlesen zu können, muss im Input Window definiert werden, welche Spalte welche Information der Untersuchung beinhaltet. Insgesamt enthält das Input Window jedoch nicht nur den Pfad zu den Daten der Untersuchung, sondern auch alle Informationen und Befehle, welche benötigt werden, um die Analyse durchführen zu können. Dies sind im Einzelnen:

1. Der Name der Analyse
2. Der Pfad zu der Datei mit den Rohdaten der Untersuchung
3. Die Spezifikation der Spalten, in welchen die Codes der beiden Rater definiert sind
4. Die Spezifikation der Spalten, in welchen die OP und TF Ratings der beiden Beurteiler aufgeführt sind
5. Die Spezifikation der Labels der einzelnen Ratings (G bis L) und eine entsprechende Skalierung (0 bis 5)
6. Die Bezeichnungen der drei Parameter mithilfe derer die Messungen modelliert werden sollen (in diesem Fall: Rater Harshness, Criteria Difficulty, Step Structure of Responses)
7. Spezifikationen über die Schätzung des Modells, sowie die Definition der Datei, in welcher die Ergebnisse gespeichert werden sollen
8. Die Definition einer Datei, in welcher die klassische Itemanalyse gespeichert werden soll

In den Output Dateien werden die Schätzungen der einzelnen Parameter, die Schätzfehler, sowie die gewichteten und ungewichteten Fit-Statistiken inklusive korrespondierender t-Werte aufgeführt. Zudem kann eine Übersicht der Verteilung der Schätzungen für alle drei Parameter eingesehen werden. Eine detaillierte Beschreibung des Vorgehens zur Analyse von Beurteilungsgeffekten findet sich ebenfalls im Manual von ACER ConQuest (Wu et al., 2007). In den Output-Dateien werden die Schätzungen der einzelnen Parameter, die Schätzfehler, sowie die gewichteten und ungewichteten Fit-Statistiken inklusive korrespondierender t-Werte aufgeführt. Zudem kann eine Übersicht der Verteilung der Scätzungen für alle drei Parameter eingesehen werden. Eine detaillierte Beschreibung des Vorgehens zur Analyse von Beurteilungsgeffekten findet sich ebenfalls im Manual von ACER ConQuest (Wu et al., 2007).

7.3.3 Beispiel 3: Mehrdimensionale latente Regression

In diesem Beispiel wollen wir die wahre Stärke von ConQuest veranschaulichen, nämlich die Möglichkeit Modelle mit mehrdimensionaler Traitstruktur zu schätzen. Dies ist der große Unterschied zu anderen Programmen, die meist nur ein- oder maximal zweidimensionale Traits zulassen. Wir verwenden die Daten von Adams et al. (1991) um eine fünfdimensionale multiple latente Regression zu schätzen. Es handelt sich um einen Datensatz, der zur Feststellung naturwissenschaftlichen Wissens von Schülern im australischen Bundesstaat Victoria erhoben wurde. Mittels einer Testbatterie, bestehend aus 65 dichotom verrechneten Multiple-Choice-Items und 4 Tests mit offenen Fragen (jeweils 9),

die polytom verrechnet worden sind, wurde das Wissen und Können der Schülerinnen anhand einiger gemeinhin als Teilbereiche der Naturwissenschaft gesehener Gebiete (offenen Fragen) und allgemeine Intelligenz im Sinne eines Matrizentests (Multiple-Choice) zu erfassen versucht. Wir werden uns fünf dieser Gebiete genauer ansehen. Jedes Wissensgebiet entspricht dabei einer latenten Dimension, das heisst, dass die Schülerinnen jeweils einen Wert auf jedem dieser latenten Kontinua zugeschrieben bekommen, je nachdem wie ausgeprägt deren Wissenstand ist. Diese latenten Dimensionen können sowohl miteinander korreliert als auch unabhängig voneinander sein. Die Autoren benannten die Dimensionen mit "Force & Motion", "Light & Sight", "Matter", "Earth & Space" und "Multiple Choice". Die mit den jeweiligen Dimensionen in Zusammenhang stehenden Personenparameter sollen dann jeweils auf die Variablen "sozioökonomischer Status" (SES), "Geschlecht" und "Schulstufe" regressiert werden, es handelt sich also um eine addititive Zerlegung der Personenfähigkeiten in drei erklärende Variablen.

Insgesamt standen zur Schätzung 2564 Schülerinnen zur Verfügung. Zuerst wollen wir zeigen wie mit ConQuest ein unbedingtes, d.h. nicht regressiertes fünfdimensionales Modell geschätzt werden kann um danach das bedingte latente Regressionsmodell zu schätzen. Mit einer unbedingten Analyse zu beginnen hat zwei Vorteile, zum Einen kann eine Art durch die Regressoren erklärte Varianz berechnet werden, zum Anderen können auf diese Weise Startwerte zur Berechung des bedingten Modells gewonnen werden. Grundsätzlich ist zu sagen, dass die Berechnung mehrdimensionaler IRT Modelle eine zeit- und rechenintensive, jedoch relativ ungenaue Sache ist. ConQuest berechnet die Parameter des Modells in diesem Fall mittels Monte-Carlo-Integration für die eine ausreichende Zahl an Stützstellen verwendet werden muss. Die Schätzung des unbedingten Modells wird somit in zwei Stufen geschehen, zuerst wird eine kleine Anzahl an Stützstellen mit einem moderaten Genauigkeitskriterium vorgegeben um sinnvolle und hoffentlich nahe an den Optima liegende Startwerte zu bekommen. Diese werden dann einer weiteren Schätzung übergeben, für die mehr Stützstellen und ein strengeres Genauigkeitskriterium verwendet werden. Die erste Stufe findet mit 400 Stützstellen statt und benötigt ca. 50 Iterationen benötigen. Die damit erhaltenen neuen Werte werden in einer Monte-Carlo-Integration mit 2000 Stützstellen feingeschliffen, was bei einem Genauigkeitskriterium von 0.002 für die Parameterschätzungen 15 Iterationen benötigt. Falls die zweite Stufe ohne den Startwerten der ersten Stufe durchgeführt wird, benötigt der Algorithmus 1000 Durchgänge. Die benötigten Befehle und das Datenfile werden in ConQuest mitgeliefert, es sei hierfür auf das Manual (Wu et al., 2007) verwiesen. Die grundsätzliche Besonderheit ist es für das benötigte Modell (hier ein PCM)

```
model tasks+tasks*steps;
```

für die zweistufige Modelschätzung

```
estimate!fit=no,method=montecarlo,nodes=400, converge=0.01;
estimate!method=montecarlo, nodes=2000, converge=0.002;
```

und für das unbedingte Modell noch

```
regression grade, gender, ses;
```

zu spezifizieren. Genauere Erklärungen finden sich im Manual.

Die Ergebnisse des unbedingten Modells sind in Abbildung 7.12 zu sehen. ConQuest schätzt zum Einen die Mittelwerte auf den einzelnen Dimensionen, zu finden im Abschnitt "Regression Coefficients" und zum Anderen die Varianz-Kovarianzmatrix der fünf latenten Dimensionen ("Covariance/Correlation Matrix"). Dies deshalb, da bei ConQuest die Annahme getroffen wird, dass sich die latenten Traits multivariat normal verteilen. In unserem Fall bedeutet dies es gibt fünf Mittelwerte, fünf Varianzen und zehn Kovarianzen/Korrelationen, die geschätzt werden müssen. Der Mittelwerte ist -0.383 für "Force & Motion", -0.905 für "Light & Sight", -0.101 für "Matter", -0.971 für die vierte Dimension "Earth & Space" und schliesslich 0.587 für die Multiple-Choice-Dimension. Die marginale Varianz wurde geschätzt als 0.630 für "Force & Motion", 0.635 für "Light & Sight", 0.656 für "Matter", 0.417 für die vierte Dimension "Earth & Space" und schliesslich 1.169 für die Multiple-Choice-Dimension. Die einzelnen Kovarianzen zwischen den latenten Dimensionen sind in 7.12 in der unteren Dreiecksmatrix gegeben, die Korrelationen in der oberen Dreiecksmatrix.

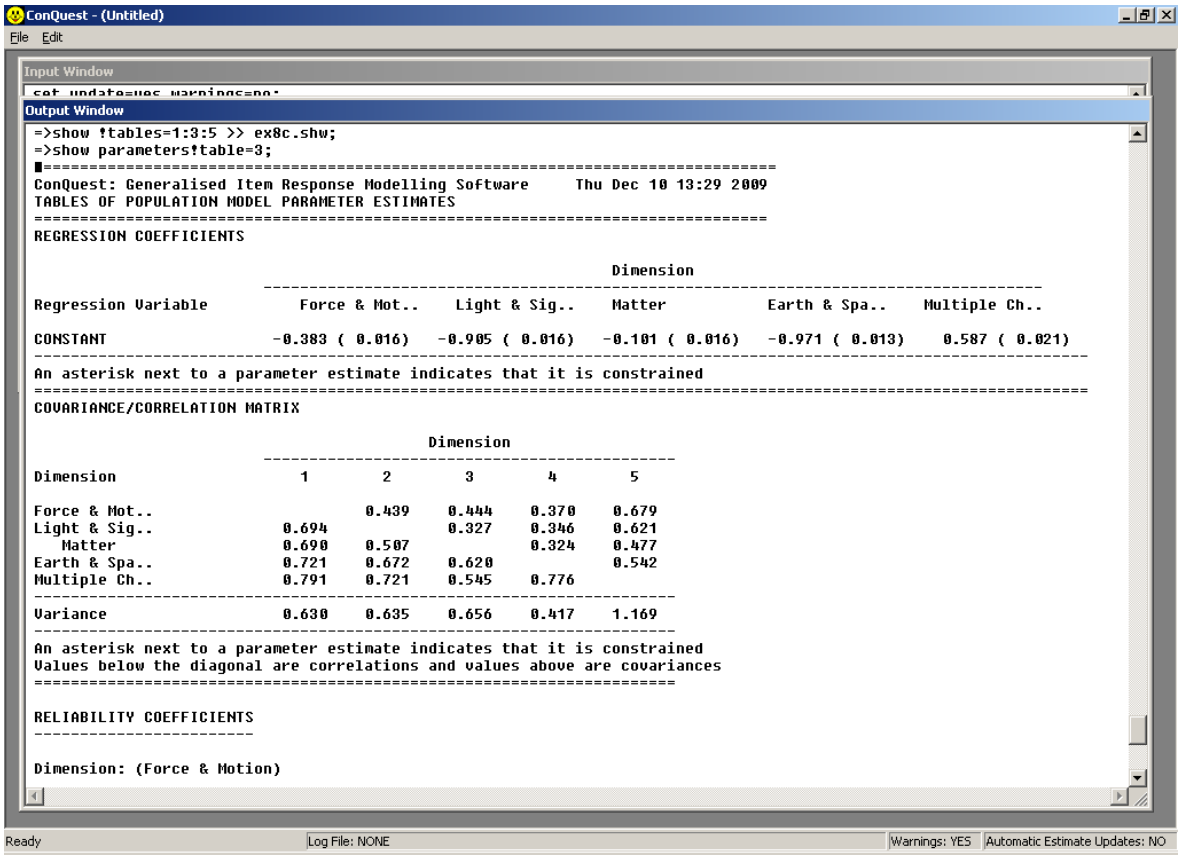

ConQuest - (Untitled)
File Edit
Input Window
Output Window
=>show !tables=1:3:5 >> ex8c.shw;
=>show parameters!table=3;
ConQuest: Generalised Item Response Modelling Software Thu Dec 10 13:29 2009
TABLES OF POPULATION MODEL PARAMETER ESTIMATES
REGRESSION COEFFICIENTS
Dimension
Regression Variable Force & Mot.. Light & Sig.. Matter Earth & Spa.. Multiple Ch..
CONSTANT -0.383 (0.016) -0.905 (0.016) -0.101 (0.016) -0.971 (0.013) 0.587 (0.021)
An asterisk next to a parameter estimate indicates that it is constrained
COVARIANCE/CORRELATION MATRIX
Dimension
Dimension 1 2 3 4 5
Force & Mot.. 0.439 0.444 0.370 0.679
Light & Sig.. 0.694 0.327 0.346 0.621
Matter 0.690 0.507 0.324 0.477
Earth & Spa.. 0.721 0.672 0.620 0.542
Multiple Ch.. 0.791 0.721 0.545 0.776
Variance 0.630 0.635 0.656 0.417 1.169
An asterisk next to a parameter estimate indicates that it is constrained
Values below the diagonal are correlations and values above are covariances
RELIABILITY COEFFICIENTS
Dimension: (Force & Motion)
Ready
Log File: NONE
Warnings: YES
Automatic Estimate Updates: NO

Abb. 7.12 ConQuest Resultat für das fünfdimensionale unbedingte Modell.

Man kann dabei sehen, dass alle latenten Traits moderat miteinander korreliert sind, mit geschätzten Korrelationskoeffizienten von 0.32-0.68. Eventuell lässt sich diese Korrelation durch die drei Regressorvariablen im unbedingten Modell erklären.

Nachdem wir diese Ergebnisse erhalten haben, wollen wir nun das Regressionmodell schätzen. Dafür verwenden wir wiederum eine Monte-Carlo-Integration mit 2000 Stützstellen und einer Scätzgenauigkeit von 0.002. Insgesamt werden dafür 29 Iterationen gebraucht, sofern die Startwerte des unbedingten Modells verwendet werden. Die Ergebnisse des unbedingten Modells sind in Abbildung 7.13 zu finden. Da wir diesmal ein Regressionsmodell für die Werte auf den latenten Dimensionen aufstellten, bekommen wir Schätzungen für die Regressionskoeffizienten für jede einzelne Dimension. So haben die drei erklärenden Variablen , "sozioökonomischer Status", "Geschlecht" und "Schulstufe" unterschiedliche Effekte auf die Populationsschätzungen. Für die Dimension "Force & Motion" wurde eine Konstante von -1.477 ermittelt. Schülerinnen aus höheren Schulstufen haben höhere Werte was Wissen in diesem Fachgebiet anbetrifft (0.737). Auch höherer sozioökonomischer Status geht mit höherem Wissensstand einher (0.365). Ebenfalls einen Effekt, wenn auch nicht besonders gross, zeigt die Variable "Geschlecht". Hierbei zeigt sich bei Jungen ein höherer Wissensstand als bei den Mädchen. Alle Effekte sind signifikant auf einem 5%-Niveau. Für die Dimension "Light & Sight" wurde eine Konstante von -1.962 ermittelt. Schülerinnen aus höheren Schulstufen haben höhere Werte was Wissen in diesem Fachgebiet anbetrifft (0.71). Auch höherer sozioökonomischer Status geht mit höherem Wissensstand einher (0.259). Auch hier zeigt die Variable "Geschlecht"einen signifikanten aber praktisch geringen Effekt. Wiederum haben Jungen ein höheren Wissensstand als die Mädchen (0.079). Alle Effekte sind signifikant auf einem 5%-Niveau. Auf der dritten Dimension "Matter" zeigt sich dasselbe Bild wie bisher. Es wurde eine Konstante von -0.920 ermittelt. Schülerinnen aus höheren Schulstufen haben höhere Werte was Wissen in diesem Fachgebiet anbetrifft (0.55). Auch höherer sozioökonomischer Status geht mit höherem Wissensstand einher (0.262). Auch hier zeigt die Variable "Geschlecht"einen knapp signifikanten aber praktisch geringen Effekt. Wiederum haben Jungen ein höheren Wissensstand als die Mädchen (0.065). Alle Effekte sind signifikant auf einem

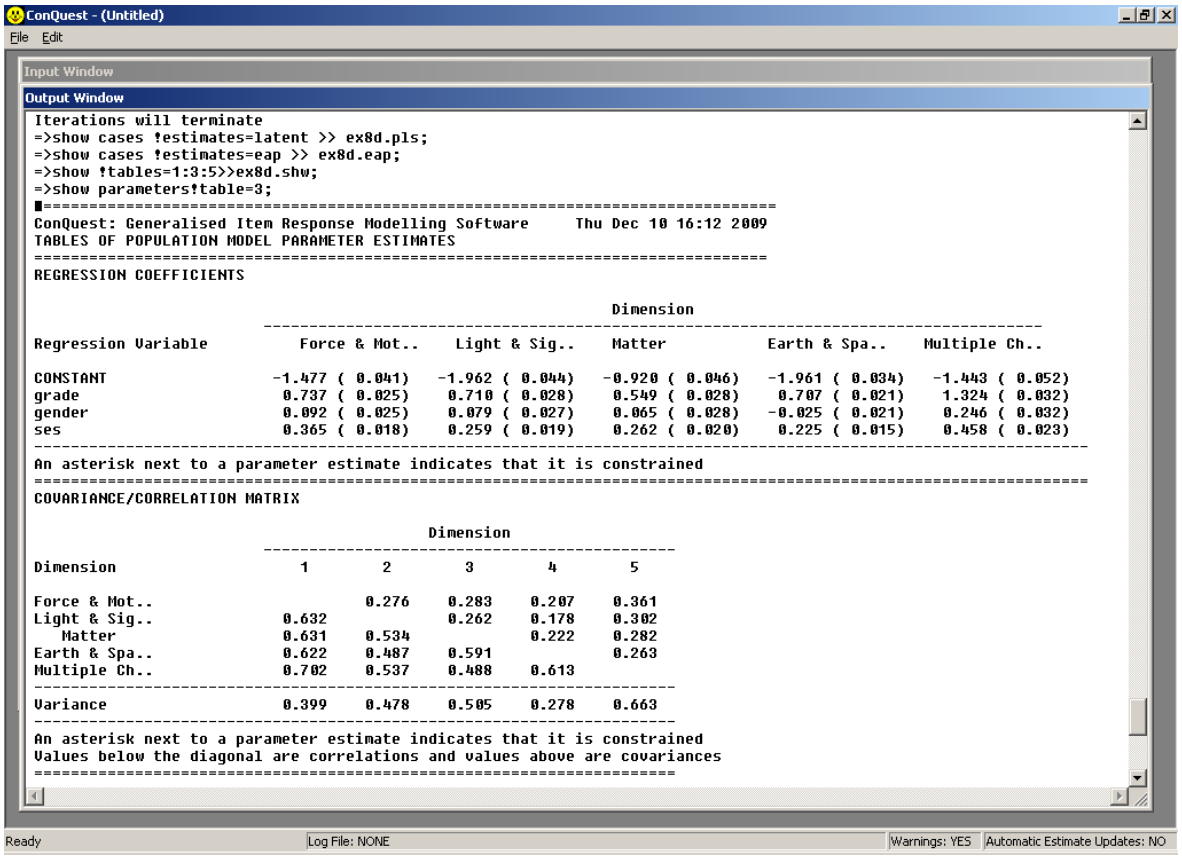

ConQuest - (Untitled)
File Edit
Input Window
Output Window
Iterations will terminate
=>show cases !estimates=latent >> ex8d.pls;
=>show cases !estimates=eap >> ex8d.eap;
=>show !tables=1:3:5>>ex8d.shw;
=>show parameters!table=3;
ConQuest: Generalised Item Response Modelling Software Thu Dec 10 16:12 2009
TABLES OF POPULATION MODEL PARAMETER ESTIMATES
REGRESSION COEFFICIENTS
Dimension
Regression Variable Force & Mot.. Light & Sig.. Matter Earth & Spa.. Multiple Ch..
CONSTANT -1.477 (0.041) -1.962 (0.044) -0.920 (0.046) -1.961 (0.034) -1.443 (0.052)
grade 0.737 (0.025) 0.710 (0.028) 0.549 (0.028) 0.707 (0.021) 1.324 (0.032)
gender 0.092 (0.025) 0.079 (0.027) 0.065 (0.028) -0.025 (0.021) 0.246 (0.032)
ses 0.365 (0.018) 0.259 (0.019) 0.262 (0.020) 0.225 (0.015) 0.458 (0.023)
An asterisk next to a parameter estimate indicates that it is constrained
COVARIANCE/CORRELATION MATRIX
Dimension
Dimension 1 2 3 4 5
Force & Mot.. 0.276 0.283 0.207 0.361
Light & Sig.. 0.632 0.262 0.178 0.302
Matter 0.631 0.534 0.222 0.282
Earth & Spa.. 0.622 0.487 0.591 0.263
Multiple Ch.. 0.702 0.537 0.488 0.613
Variance 0.399 0.478 0.505 0.278 0.663
An asterisk next to a parameter estimate indicates that it is constrained
Values below the diagonal are correlations and values above are covariances
Ready
Log File: NONE
Warnings: YES
Automatic Estimate Updates: NO

Abb. 7.13 ConQuest Resultat für das fünfdimensionale bedingte Regressionsmodell mit den erklärenden Variablen SES, Geschlecht (Gender) und Schulstufe ("Grade").

5%-Niveau. Für die vierte Dimension "Earth & Space" zeigt sich ein ähnliches Bild wie bisher, mit einer Ausnahme. Diesmal fällt es den Mädchen leichter, die Fragen zu beantworten (-0.025). Dieser Effekt ist jedoch nicht signifikant auf einem 5%-Niveau, weswegen er auch genauso gut zufällig zu Stande kommen könnte. Ansonsten wurde eine Konstante von -1.961 ermittelt, dieses scheint also das schwerste Fachgebiet insgesamt gewesen zu sein. Schülerinnen aus höheren Schulstufen haben höhere Werte was Wissen in diesem Fachgebiet anbetrifft (0.71). Auch höherer sozioökonomischer Status geht mit höherem Wissensstand einher (0.23). Die letzten drei Effekte sind signifikant auf einem 5%-Niveau. Die fünfte Dimension schliesslich, "Multiple-Choice" weist wiederum das alt bekannte Muster auf. Schülerinnen aus höheren Schulstufen haben höhere Werte erreicht (1.32), höherer sozioökonomischer Status geht mit höheren Punktzahlen einher (0.46) und Jungen erreichen höhere Punktzahlen als Mädchen. Die Konstante wurde mit -1.443 geschätzt. Wiederum sind alle Effekte signifikant auf einem 5%-Niveau. Für die marginalen Varianzen wurden 0.399 für "Force & Motion", 0.48 für "Light & Sight", 0.51 für "Matter", 0.28 für die vierte Dimension "Earth & Space" und schliesslich 0.66 für die Multiple-Choice-Dimension.

Der zusätzlich durch die Regressanden eingebrachte Erklärungsgehalt reduziert die Varianz um einiges. Der Anteil an durch die Regressoren erklärter Varianz des unbedingten Modells kann mit

$$R^2 = 1 - \frac{VAR_{cond}}{VAR_{uncond}}, \tag{7.14}$$

berechnet werden. Hierbei steht VAR_{uncond} für die geschätzte Varianz der jeweiligen latenten Dimension für das unbedingte Modell und VAR_{cond} dieselbige für das Regressionsmodell. Für die erste Dimension, "Force & Motion" ist der erklärte Anteil 34%, die unbedingte Varianz auf der zweiten Dimension "Light & Sight" wird zu 21% durch die Regressoren erklärt. "Matter" erfährt eine Reduktion auf 88%, d.h. 12% erklärter Varianzanteil. Für "Earth & Space" ist dieser Anteil 36% und bei der fünften und letzten Dimension, "Multiple-Choice" führen die erklärenden Variablen

sogar zu einem 45% Anteil erklärter Varianz. Die einzelnen Kovarianzen zwischen den latenten Dimensionen sind in 7.13 in der unteren Dreiecksmatrix gegeben, die Korrelationen in der oberen Dreiecksmatrix. Man kann dabei sehen, dass sich bei allen Traits die bivariaten Korrelationen reduziert haben und jetzt relativ gering sind (0.18-0.36). Der zusätzliche Erklärungsgehalt, den die Regressoren miteinbrachten, kann zumindest etwas der wechselseitigen Korrelation erklären. Trotzdem ist noch substanzielle Restkorrelation vorhanden, man kann also nicht von bedingt auf SES, Geschlecht und Schulstufe unabhängigen Dimensionen sprechen. Sie sind nach wie vor konfudiert. Tabelle 7.1 gibt eine kurze Zusammenfassung der Analyse wieder. In der ersten Zeile findet sich der Anteil erklärter Varianz des unbedingten Modells aufgrund der Regressoren wieder, berechnet als $1 - VAR_{cond}/VAR_{uncond}$, wobei VAR_{uncond} die geschätzte Varianz der jeweiligen latenten Dimension für das unbedingte Modell beschreibt und VAR_{cond} dieselbige für das Regressionsmodell. In den Zeilen 2-4 sind die entsprechenden Punktschätzer der Regressionskoeffizienten für die einzelnen Dimensionen angeführt.

Schätzung	Force & Motion	Light& Sight	Matter	Earth& Space	Multiple Choice
Erkl. Varianz	34%	21%	12%	36%	45%
Schulstufe	1.188	1.134	0.900	0.586	1.143
Geschlecht	0.160	0.125	0.083	-0.039	0.213
SES	0.600	0.425	0.417	0.511	0.396

Tabelle 7.1 Zusammenfassung des fünfdimensionalen latenten Regressionsmodell

Kapitel 8
IRT Analysis using the `ltm` Package

Dimitris Rizopoulos

8.1 Basics of IRT

Item Response Theory (IRT) (Baker und Kim, 2004; van der Linden und Hambleton, 1997) considers a class of latent variable models that link mainly dichotomous and polytomous manifest (i.e., response) variables to a set of latent variables. The main applications of IRT can be found in educational testing in which analysts are interested in measuring examinees' ability using a test that consists of several items (i.e., questions). Several models and estimation procedures have been proposed that deal with various aspects of educational testing.

The aim of this document is to present the R package `ltm`, available from CRAN (http://cran.r-project.org), that can be used for latent variable modelling under the IRT approach. For a more complete list of the available CRAN packages relevant to the analysis of event time data we refer to the CRAN Task View http://cran.r-project.org/web/views/Psychometrics.html

The basic idea of latent variable analysis is to find, for a given set of response variables $x_1,\ldots,x_p$, a set of latent variables $z_1,\ldots,z_q$ (with $q \ll p$) that contains essentially the same information about dependence. The latent variable regression models have usually the following form

$$E(x_i \mid z) = g(\lambda_{i0} + \lambda_{i1} z_1 + \cdots + \lambda_{iq} z_q),\ i = 1,\ldots,p, \tag{8.1}$$

where $g(\cdot)$ is a link function, $\lambda_{i0},\ldots,\lambda_{iq}$ are the regression coefficients for the ith manifest variable, and x_i is independent of x_j, for $i \neq j$, given $z = \{z_1,\ldots,z_q\}$. The common factor analysis model assumes that the x_i's are continuous random variables following a Normal distribution with $g(\cdot)$ being the identity link. In this document we primarily focus on IRT models, and consider mainly dichotomous and polytomous items, in which $E(x_i \mid z)$ expresses the probability of endorsing one of the possible response categories. In the IRT framework usually a single latent variable is assumed to capture the interrelationships between the manifest variables; however, several extensions have been proposed that consider multidimensional and/or hierarchical latent variable designs (Doran et al., 2007; Fox, 2007).

8.1.1 IRT Models for Dichotomous Data

The basic ingredient of the IRT modelling for dichotomous data is the model for the probability of positive (or correct) response in each item given the ability level z. A general model for this probability for the mth examinee in the ith item is the following

$$\Pr(x_{im} = 1 \mid z_m) = c_i + (1 - c_i) g\{\alpha_i (z_m - \beta_i)\}, \tag{8.2}$$

Ursprünglich erschienen in: Rizopoulos (2006)
Erasmus University Medical Center, Department of Biostatistics • d.rizopoulos@erasmusmc.nl

where x_{im} is the dichotomous manifest variable, z_m denotes the examinee's level on the latent scale, c_i is the guessing parameter, α_i the discrimination parameter and β_i the difficulty parameter. The guessing parameter expresses the probability an examinee with very low ability to respond correctly to an item by chance. The discrimination parameter quantifies how well the item distinguishes between subjects with low/high standing in the latent scale, and the difficulty parameter expresses the difficulty level of the item.

The one-parameter logistic model, also known as the Rasch model (Rasch, 1960), assumes that there is no guessing parameter, i.e., $c_i = 0$ and that the discrimination parameter equals one, i.e., $\alpha_i = 1$, $\forall i$. The two-parameter logistic model allows for different discrimination parameter per item and assumes that $c_i = 0$. Finally, the Birnbaum's three-parameter model (Birnbaum, 1968) estimates all three parameters per item.

The two most common choices for $g(\cdot)$ are the probit and the logit link, that correspond to the cumulative distribution function (cdf) of the normal and logistic distributions, respectively. The functions included in `ltm` fit (8.2) under the logit link. Approximate results under the probit link for the one- and two-parameter logistic models can be obtained using the relation

$$\alpha_i^{(p)}(z_m - \beta_i^{(p)}) \approx 1.702\alpha_i^{(l)}(z_m - \beta_i^{(l)}),$$

where $\alpha_i^{(p)}$, $\alpha_i^{(l)}$ are the discrimination parameters under the probit and logit link, respectively, and $\beta_i^{(p)}$, $\beta_i^{(l)}$ are defined analogously. The scaling constant 1.702 is chosen such that the absolute difference between the normal and logistic cdf is less than 0.01 over the real line.

8.1.2 IRT Models for Polytomous Data

Analysis of polytomous manifest variables is currently handled by `ltm` using the Graded Response Model (GRM) and the Generalized Partial Credit Model (GPCM). The GRM was first introduced by Samejima (1969), and postulates that the probability of the mth subject to endorse the kth response for the ith item is expressed as

$$\Pr(x_{im} = k \mid z_m) = g(\eta_{ik}) - g(\eta_{i,k+1}), \tag{8.3}$$

$$\eta_{ik} = \alpha_i(z_m - \beta_{ik}), \quad k = 1, \ldots, K_i,$$

where x_{im} is the ordinal manifest variable with K_i possible response categories, z_m is the standing of the mth subject in the latent trait continuum, α_i denotes the discrimination parameter, and β_{ik}'s are the extremity parameters with $\beta_{i1} < \ldots < \beta_{ik} < \ldots < \beta_{i,K_i-1}$ and $\beta_{iK_i} = \infty$. The interpretation of α_i is essentially the same as in the models for dichotomous data. However, in GRM the β_{ik}'s represent the cut-off points in the cumulative probabilities scale and thus their interpretation is not direct – `ltm` fits the GRM under the logit link.

The GPCM was first discussed by Masters (1982) and it was later extended by Muraki (1992):

$$\Pr(x_{im} = k \mid z_m) = \frac{\exp \sum_{c=0}^{k} \alpha_i(z_m - \beta_{ic})}{\sum_{r=0}^{K_i} \exp \sum_{c=0}^{r} \alpha_i(z_m - \beta_{ic})}, \tag{8.4}$$

where $\sum_{c=0}^{0} \beta_i(z - \beta_{ic}) \equiv 0$.

8.1.3 Marginal Maximum Likelihood Estimation

Package ltm estimates the models parameters in the above defined models using a Marginal Maximum Likelihood approach. In particular, we assume that the subjects represent a random sample from a population and their ability is distributed according to a distribution function $F(z)$. The model parameters are estimated by maximizing the observed data log-likelihood obtained by integrating out the latent variables; the contribution of the mth sample unit is

$$\ell_m(\theta) = \log p(x_m;\theta) = \log \int p(x_m|z_m;\theta)\, p(z_m)\, dz_m, \tag{8.5}$$

where $p(\cdot)$ denotes a probability density function, x_m denotes the vector of responses for the mth sample unit, z_m is assumed to follow a standard normal distribution and $\theta = (\alpha_i, \beta_i)$. The integral in (8.5) is approximated using the Gauss-Hermite quadrature rule. It is known (Pinheiro und Bates, 1995) that the number of quadrature points used may influence the parameter estimates. This is especially the case for more complex latent variable models that assume more than one latent variable and/or hierarchical designs. For unidimensional IRT models considered the default number of points will be, in the majority of the cases, sufficient.

8.1.4 Estimating the Latent Variable

The factor scores are usually calculated as the mean of the posterior distribution

$$\tilde{z}_m = \int z_m p(z_m \mid x_m;\theta)\, dz_m, \tag{8.6}$$

or the mode of the posterior distribution

$$\hat{z}_m = \arg\max_z \{p(x_m|z_m;\theta)p(z_m)\}, \tag{8.7}$$

for each sample unit. The posterior means (8.6) can be calculated using the Gauss-Hermite quadrature rule, whereas for the posterior modes (8.7) a numerical optimizer such as is optim() required. Note that in (8.7) we typically replace the true parameter values θ by their maximum likelihood estimate $\hat{\theta}$. Thus, in small samples we in fact ignore the variability of plugging-in estimates instead of the true parameter values. To take this into account, Rizopoulos und Moustaki (2008) have proposed a Multiple Imputation like approach in which the uncertainty about the true parameter values is explicitly acknowledged.

8.2 IRT Analysis for Dichotomous Data

To illustrate the capabilities of the R package ltm for IRT analysis of dichotomous items, we will present a sample analysis for the Sexual Attitudes data set. This data set is extracted from the 1990 British Social Attitudes Survey, and concerns contemporary sexual attitudes. The questions addressed to 1077 individuals were as follows:

1. Should divorce be easier?
2. Do you support the law against sexual discrimination?
3. View on pre-marital sex: (wrong/not wrong)
4. View on extra-marital sex: (wrong/not wrong)
5. View on sexual relationship between individuals of the same sex: (wrong/not wrong)
6. Should gays teach in school?
7. Should gays teach in higher education?

8. Should gays hold public positions?
9. Should a female homosexual couple be allowed to adopt children?
10. Should a male homosexual couple be allowed to adopt children?

More information regarding this study can be found at: http://www.britsocat.com/. A copy of the data set that we are going to use here for our illustrations, can be downloaded from the following website (we directly import the data in R using `read.table()`):

```
R> SexAtt <- read.table("http://eur.academia.edu/documents/0009/0840/sexualat.dat")
R> names(SexAtt) <- c("divorce", "sexdisc", "premar", "exmar",
+ "gaysex", "gayscho", "gayhied", "gaypubl", "gayfadop", "gaymadop")
R> head(SexAtt)

  divorce sexdisc premar exmar gaysex gayscho gayhied gaypubl gayfadop gaymadop
1       0       1      1     0      0       1       1       1        0        0
2       0       1      1     0      0       1       1       1        0        0
3       0       1      1     0      0       1       1       0        0        0
4       0       1      1     0      0       1       1       1        0        0
5       0       1      1     0      1       1       1       0        0        0
6       0       1      1     0      0       0       0       0        0        0
```

8.2.1 Descriptive Analysis

We will start with a descriptive analysis of the data, with descriptive measures relevant to test theory. These measures are calculated by function `descript()`:

```
R> library(ltm)

This is package 'ltm' version '0.9-4'

R> dsc <- descript(SexAtt)
R> dsc

Descriptive statistics for the 'SexAtt' data-set

Sample:
 10 items and 1077 sample units; 0 missing values

Proportions for each level of response:
              0      1   logit
divorce  0.8737 0.1263 -1.9343
sexdisc  0.1755 0.8245  1.5472
premar   0.2331 0.7669  1.1911
exmar    0.8700 0.1300 -1.9010
gaysex   0.7112 0.2888 -0.9014
gayscho  0.5246 0.4754 -0.0985
gayhied  0.4522 0.5478  0.1919
gaypubl  0.4113 0.5887  0.3585
gayfadop 0.8143 0.1857 -1.4782
gaymadop 0.8942 0.1058 -2.1339


Frequencies of total scores:
      0   1   2   3   4   5   6  7  8  9 10
Freq 29 145 179 126 126 158 159 60 58 32  5


Point Biserial correlation with Total Score:
         Included Excluded
divorce    0.1805   0.0387
sexdisc    0.3156   0.1587
premar     0.5039   0.3499
```

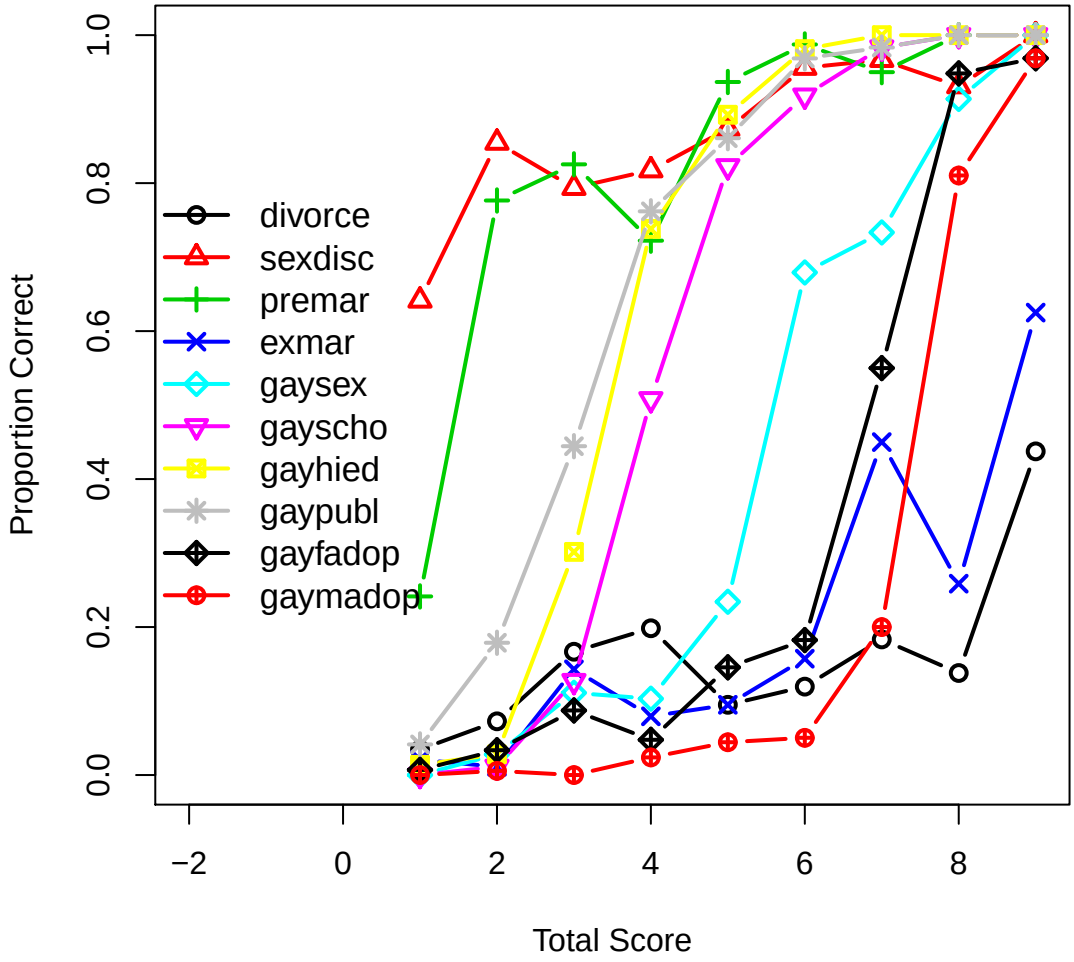
```
exmar      0.3545   0.2194
gaysex     0.6595   0.5262
gayscho    0.7818   0.6733
gayhied    0.7820   0.6740
gaypubl    0.7118   0.5806
gayfadop   0.5642   0.4338
gaymadop   0.5506   0.4484


Cronbach's alpha:
                    value
All Items          0.7558
Excluding divorce  0.7773
Excluding sexdisc  0.7679
Excluding premar   0.7453
Excluding exmar    0.7588
Excluding gaysex   0.7190
Excluding gayscho  0.6915
Excluding gayhied  0.6915
Excluding gaypubl  0.7088
Excluding gayfadop 0.7338
Excluding gaymadop 0.7349


Pairwise Associations:
   Item i Item j p.value
1       1      8   0.917
2       1      7   0.853
3       1      6   0.692
4       1      5   0.652
5       2      9   0.459
6       1      2   0.264
7       2      4   0.148
8       2     10   0.090
9       1      9   0.087
10      1     10   0.034
```

This function returns among others, the percentages of positive and negative responses for each item, biserial correlations of each item with the total score (i.e., the sum of positive responses of each subject), Cronbach's α for all items and excluding one item each time, and the χ^2 p-values for pairwise associations between the ten items, corresponding to the 2×2 contingency tables for all possible pairs. Inspection of non significant results can be used to reveal 'problematic' items[1].

Function `descript()` returns objects of class 'descript' for which a `plot()` method is available that produces the scatterplot of the proportion of positive responses for each item versus the total score

```
R> plot(dsc, type = "b", lty = 1, pch = 1:10, col = 1:10, lwd = 2,
+     cex = 1.1, xlim = c(-2, 9))
R> legend("left", names(SexAtt), pch = 1:10, col = 1:10, lty = 1, lwd = 2,
+     cex = 1.1, bty = "n")
```

[1] Latent variable models assume that the high associations between items can be explained by a set of latent variables. Thus, for pairs of items that do not reject independence we could say that they violate this assumption.



8.2.2 IRT Models for Dichotomous Data

8.2.2.1 Fitting

We initially fit the original form of the Rasch model that assumes a known discrimination parameter fixed at one. The version of the Rasch model fitted by `rasch()` in `ltm` assumes equal discrimination parameters across items but by default estimates its value, i.e., for p items $\alpha_1 = \ldots = \alpha_p = \alpha$. In order to impose the constraint $\alpha = 1$, the `constraint` argument is used. This argument accepts a two-column matrix where the first column denotes the parameter and the second column indicates the value at which the corresponding parameter should be fixed. Parameters are fixed under the additive parameterization $\lambda_{io} + \lambda z_m$; for instance, for p items the numbers $1,\ldots,p$, in the first column of `constraint`, correspond to parameters $\lambda_{1o},\ldots,\lambda_{po}$, and the number $p+1$ to the discrimination parameter λ[2]. Thus, for the Sexual Attitudes data set we fix the discrimination parameter at one and fit the Rasch model by:

```
R> fit.rasch1 <- rasch(SexAtt, constraint = cbind(ncol(SexAtt) + 1, 1))
R> summary(fit.rasch1)
```

[2] Note that under both parameterizations, the discrimination parameter coincides, i.e., $\lambda \equiv \alpha$.

```
Call:
rasch(data = SexAtt, constraint = cbind(ncol(SexAtt) + 1, 1))

Model Summary:
   log.Lik      AIC      BIC
 -5030.831 10081.66 10131.48

Coefficients:
                  value std.err   z.vals
Dffclt.divorce   2.3769  0.1042  22.8117
Dffclt.sexdisc  -1.9210  0.0928 -20.6895
Dffclt.premar   -1.4958  0.0855 -17.5033
Dffclt.exmar     2.3384  0.1031  22.6721
Dffclt.gaysex    1.1353  0.0813  13.9604
Dffclt.gayscho   0.1201  0.0755   1.5906
Dffclt.gayhied  -0.2501  0.0756  -3.3075
Dffclt.gaypubl  -0.4618  0.0762  -6.0565
Dffclt.gayfadop  1.8400  0.0917  20.0694
Dffclt.gaymadop  2.6061  0.1111  23.4638
Dscrmn           1.0000      NA       NA

Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 5.7e-05
quasi-Newton: BFGS
```

The `summary()` method returns more detailed output containing the parameter estimates and standard errors. We can transform these estimates to probabilities of positive response for the average subject using the `coef()` method

```
R> # items ordered by difficulty, and
R> # probability of positive response by the average
R> # individual
R> coef(fit.rasch1, prob = TRUE, order = TRUE)

             Dffclt Dscrmn P(x=1|z=0)
sexdisc  -1.9209995      1 0.87224985
premar   -1.4958421      1 0.81695353
gaypubl  -0.4617927      1 0.61343937
gayhied  -0.2501496      1 0.56221332
gayscho   0.1200687      1 0.47001883
gaysex    1.1353410      1 0.24317679
gayfadop  1.8399917      1 0.13705227
exmar     2.3383781      1 0.08799399
divorce   2.3768954      1 0.08495159
gaymadop  2.6061011      1 0.06874680
```

The column `P(x=1|z=0)` corresponds to $P(x_i = 1 \mid z = 0)$ under (8.2), and denotes the probability of a positive response to the ith item for the average individual. The `order` argument can be used to sort the items according to the difficulty estimates. We continue by fitting the unconstrained Rasch model that estimates the discrimination parameter. This achieved using function `rasch()` but without specifying the `constraint` argument

```
R> fit.rasch2 <- rasch(SexAtt)
R> summary(fit.rasch2)

Call:
rasch(data = SexAtt)

Model Summary:
```

```
   log.Lik      AIC      BIC
 -4939.777 9901.555 9956.356

Coefficients:
                  value std.err   z.vals
Dffclt.divorce   1.6818  0.0834  20.1692
Dffclt.sexdisc  -1.3677  0.0735 -18.6102
Dffclt.premar   -1.0687  0.0664 -16.0923
Dffclt.exmar     1.6553  0.0825  20.0671
Dffclt.gaysex    0.8136  0.0618  13.1711
Dffclt.gayscho   0.0879  0.0553   1.5894
Dffclt.gayhied  -0.1784  0.0556  -3.2098
Dffclt.gaypubl  -0.3306  0.0563  -5.8675
Dffclt.gayfadop  1.3095  0.0722  18.1333
Dffclt.gaymadop  1.8392  0.0891  20.6517
Dscrmn           1.6283  0.0563  28.9460

Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 0.0067
quasi-Newton: BFGS
```

```
R> coef(fit.rasch2, prob = TRUE, order = TRUE)

              Dffclt   Dscrmn P(x=1|z=0)
sexdisc  -1.36771096 1.628349 0.90265783
premar   -1.06873897 1.628349 0.85072268
gaypubl  -0.33056611 1.628349 0.63141154
gayhied  -0.17836796 1.628349 0.57210516
gayscho   0.08792215 1.628349 0.46426902
gaysex    0.81362272 1.628349 0.21001051
gayfadop  1.30950365 1.628349 0.10599405
exmar     1.65530469 1.628349 0.06324449
divorce   1.68184333 1.628349 0.06073209
gaymadop  1.83921960 1.628349 0.04765717
```

We observe that the estimated discrimination parameter seems considerably different than one. To statistically test if the discrimination parameter is indeed different than one, we perform s likelihood ratio test using function `anova()`:

```
R> anova(fit.rasch1, fit.rasch2)

 Likelihood Ratio Table
                AIC      BIC  log.Lik    LRT df p.value
fit.rasch1 10081.66 10131.48 -5030.83
fit.rasch2  9901.55  9956.36 -4939.78 182.11  1  <0.001
```

The definitions of AIC and BIC used by the `summary()` and `anova()` methods in `ltm` are such that "smallerïs better.

In order to check the fit of the Rasch models to the data, the `GoF.rasch()` and `margins()` functions are used. The `GoF.rasch()` function performs a parametric Bootstrap goodness-of-fit test using Pearson's χ^2 statistic. In particular, the null hypothesis states that the observed data have been generated under the Rasch model with parameter values the maximum likelihood estimates $\hat{\theta}$. To test this hypothesis B samples are generated under the Rasch model using $\hat{\theta}$, and the Pearson's χ^2 statistic T_b $(b = 1,\ldots,B)$ is computed for each data-set; the p-value is then approximated by the number of times $T_b \geq T_{obs}$ plus one, divided by $B+1$, where T_{obs} denotes the value of the statistic in the original data-set. For the Sexual Attitudes data set this procedure yields:

```
R> set.seed(1234)
R> GoF.rasch(fit.rasch1, B = 200)
```

```
Bootstrap Goodness-of-Fit using Pearson chi-squared

Call:
rasch(data = SexAtt, constraint = cbind(ncol(SexAtt) + 1, 1))

Tobs: 4374.69
# data-sets: 201
p-value: 0.005
```

```
R> GoF.rasch(fit.rasch2, B = 200)
```

```
Bootstrap Goodness-of-Fit using Pearson chi-squared

Call:
rasch(data = SexAtt)

Tobs: 6441.21
# data-sets: 201
p-value: 0.005
```

An alternative method to investigate the fit of the model is to examine the two- and three-way χ^2 residuals produced by the `margins()` method (we illustrate for the unconstrained Rasch model):

```
R> # by default the two-way margins are produced
R> margins(fit.rasch2)
```

```
Call:
rasch(data = SexAtt)

Fit on the Two-Way Margins

Response: (0,0)
  Item i Item j Obs    Exp (O-E)^2/E
1      6      7 477 339.81     55.39 ***
2      7      8 382 282.62     34.94 ***
3      6      8 407 314.41     27.27 ***

Response: (1,0)
  Item i Item j Obs    Exp (O-E)^2/E
1      6      7  10 145.74    126.42 ***
2      1      2  29   5.30    106.00 ***
3      1      8  57  19.65     70.97 ***

Response: (0,1)
  Item i Item j Obs    Exp (O-E)^2/E
1      6      7  88 224.84     83.29 ***
2      9     10   2  62.18     58.24 ***
3      2      9  31   8.95     54.29 ***

Response: (1,1)
  Item i Item j Obs    Exp (O-E)^2/E
1      9     10 112  51.07     72.71 ***
2      6      7 502 366.61     50.00 ***
3      7      8 529 433.07     21.25 ***

'***' denotes a chi-squared residual greater than 3.5
```

```
R> # for the three-way margins we specify the 'type' argument
R> margins(fit.rasch2, type = "three-way")
```

```
Call:
rasch(data = SexAtt)

Fit on the Three-Way Margins

Response: (0,0,0)
```

```
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      6      7      8 379 224.07    107.12 ***
2      5      6      7 436 306.32     54.90 ***
3      6      7     10 465 332.02     53.26 ***

Response: (1,0,0)
  Item i Item j Item k Obs  Exp (O-E)^2/E
1      1      6      7  62 9.88    274.95 ***
2      1      6      8  53 8.62    228.39 ***
3      1      7      8  46 7.05    215.20 ***

Response: (0,1,0)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      1      6      7   9 132.42    115.03 ***
2      4      6      7  10 131.95    112.71 ***
3      5      6      7   9 109.19     91.93 ***

Response: (1,1,0)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      2      6      7   7 124.84    111.23 ***
2      3      6      7   8 116.34    100.90 ***
3      6      7      9 349 239.21     50.39 ***

Response: (0,0,1)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      1      6      7  76 204.30     80.57 ***
2      4      6      7  77 203.57     78.70 ***
3      5      6      7  73 168.46     54.09 ***

Response: (1,0,1)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      1      2      9  10   0.92     90.04 ***
2      1      2      3  23   4.24     83.02 ***
3      2      6      7  73 192.60     74.27 ***

Response: (0,1,1)
  Item i Item j Item k Obs   Exp (O-E)^2/E
1      2      9     10  13  0.75    199.88 ***
2      1      9     10  91 26.87    153.10 ***
3      4      9     10  79 26.43    104.56 ***

Response: (1,1,1)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      6      7      8 469 298.56     97.30 ***
2      5      9     10  89  37.60     70.28 ***
3      3      9     10 106  49.88     63.15 ***

'***' denotes a chi-squared residual greater than 3.5
```

From both goodness-of-fit methods we observe that the Rasch models do not seem to capture the interrelationships between the different items. We continue by further extending the model and namely assuming a different discrimination parameter per item. As we have seen in Section 8.1.1, this extension is known as the two parameter logistic model. This model is fitted in `ltm` using function `ltm()`. Contrary to `rasch()`, function `ltm()` accepts as first argument an R formula that in its left-hand side the `data.frame` or `matrix` of dichotomous responses is provided, and in its right-hand the latent structure is specified. For the latent structure, up to two latent variables are allowed with code names `z1` and `z2`. For the two parameter logistic model we need to specify a single latent variable as

```
R> fit.2pl <- ltm(SexAtt ~ z1)
R> summary(fit.2pl)

Call:
ltm(formula = SexAtt ~ z1)
```

```
Model Summary:
   log.Lik     AIC      BIC
 -4531.825 9103.65 9203.288

Coefficients:
                   value  std.err   z.vals
Dffclt.divorce  18.3530  17.2908   1.0614
Dffclt.sexdisc  -3.1843   0.5334  -5.9696
Dffclt.premar   -1.5166   0.1347 -11.2616
Dffclt.exmar     3.3550   0.5495   6.1051
Dffclt.gaysex    0.6687   0.0689   9.6997
Dffclt.gayscho  -0.0295   0.2951  -0.0999
Dffclt.gayhied  -0.2403   0.0737  -3.2624
Dffclt.gaypubl  -0.3990   0.0404  -9.8683
Dffclt.gayfadop  1.2327   0.0973  12.6701
Dffclt.gaymadop  1.5058   0.0933  16.1362
Dscrmn.divorce   0.1050   0.0987   1.0641
Dscrmn.sexdisc   0.5322   0.0987   5.3927
Dscrmn.premar    1.0223   0.1126   9.0756
Dscrmn.exmar     0.5869   0.1013   5.7921
Dscrmn.gaysex    1.7195   0.1421  12.0996
Dscrmn.gayscho  18.6195 185.9020   0.1002
Dscrmn.gayhied   9.0649   2.1280   4.2598
Dscrmn.gaypubl   3.2512   0.2698  12.0511
Dscrmn.gayfadop  1.5619   0.1406  11.1080
Dscrmn.gaymadop  2.3238   0.2441   9.5209

Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 0.0088
quasi-Newton: BFGS
```

The fit on both the two- and three-way margins remains problematic but it is a little bit better than the fit of the Rasch models

```
R> margins(fit.2pl)

Call:
ltm(formula = SexAtt ~ z1)

Fit on the Two-Way Margins

Response: (0,0)
  Item i Item j Obs    Exp (O-E)^2/E
1      6      7 477 399.50     15.04 ***
2      6      8 407 344.06     11.51 ***
3      7      8 382 324.47     10.20 ***

Response: (1,0)
  Item i Item j Obs    Exp (O-E)^2/E
1      9     10  88 153.29     27.81 ***
2      3      7 311 262.37      9.01 ***
3      2      8 348 298.82      8.10 ***

Response: (0,1)
  Item i Item j Obs   Exp (O-E)^2/E
1      9     10   2 58.48     54.54 ***
2      7     10  12  2.45     37.24 ***
3      6     10  15  4.73     22.33 ***
```

```
Response: (1,1)
  Item i Item j Obs    Exp (O-E)^2/E
1      9     10 112  65.66     32.71 ***
2      3      6 449 516.05      8.71 ***
3      3      7 515 582.51      7.82 ***

'***' denotes a chi-squared residual greater than 3.5
```

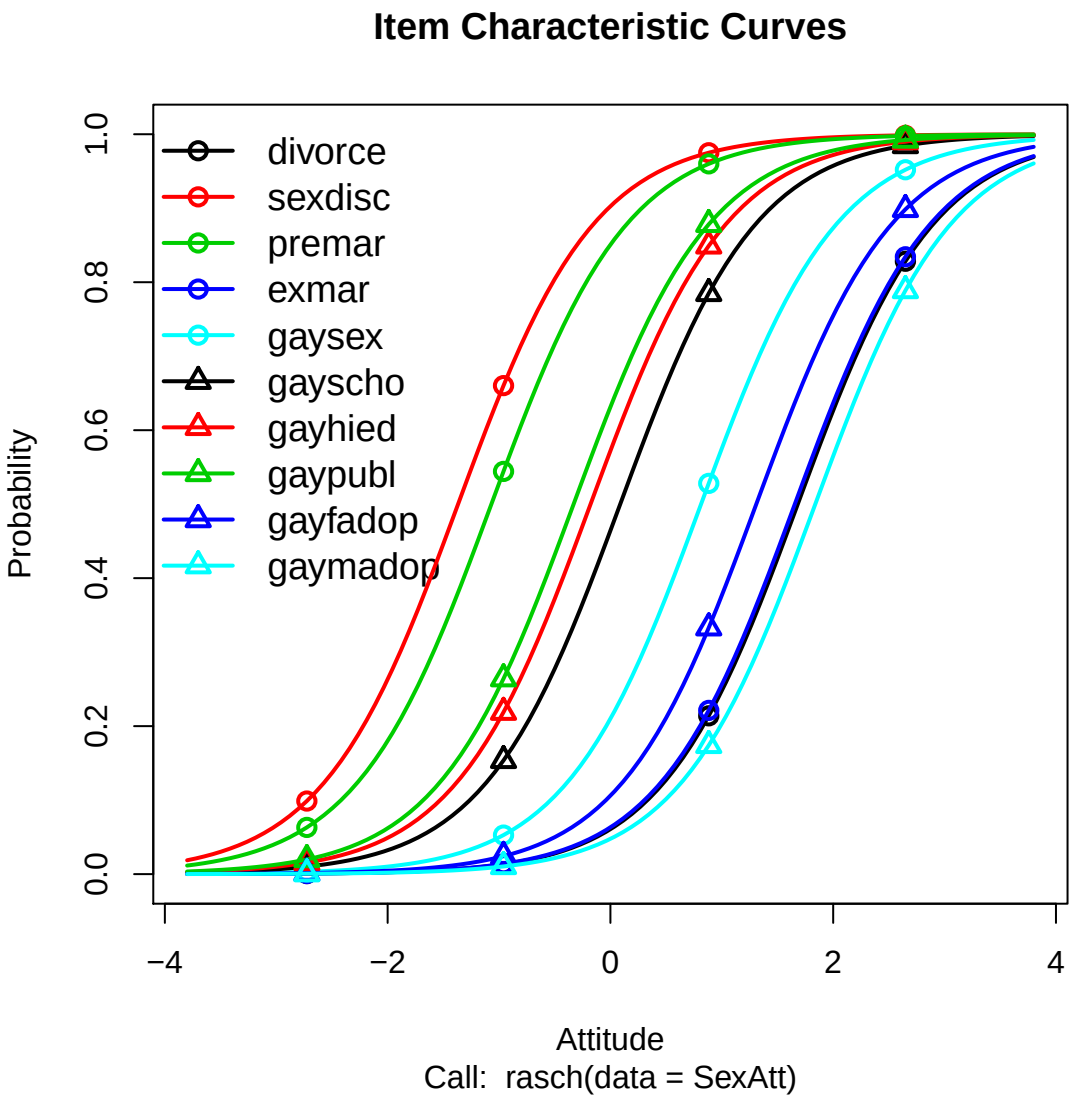
```
R> margins(fit.2pl, type = "three-way")
```

```
Call:
ltm(formula = SexAtt ~ z1)

Fit on the Three-Way Margins

Response: (0,0,0)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      3      4      5 233 178.65     16.53 ***
2      1      3      5 219 172.24     12.70 ***
3      6      7     10 465 397.61     11.42 ***

Response: (1,0,0)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      3      6      7 303 246.83     12.78 ***
2      2      6      8 314 258.97     11.69 ***
3      2      6      7 361 301.86     11.59 ***

Response: (0,1,0)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      1      9     10  76 132.65     24.19 ***
2      4      9     10  72 126.49     23.47 ***
3      3      5     10  11  32.48     14.20 ***

Response: (1,1,0)
  Item i Item j Item k Obs    Exp (O-E)^2/E
1      7      9     10  64 136.50     38.51 ***
2      6      9     10  57 126.80     38.43 ***
3      2      9     10  70 134.52     30.95 ***

Response: (0,0,1)
  Item i Item j Item k Obs   Exp (O-E)^2/E
1      7      8     10  10  1.35     55.57 ***
2      6      7     10  12  1.89     54.02 ***
3      1      9     10   1 50.34     48.36 ***

Response: (1,0,1)
  Item i Item j Item k Obs  Exp (O-E)^2/E
1      5      7     10   8 0.34    170.34 ***
2      5      6     10   9 0.89     74.32 ***
3      4      7     10   4 0.24     59.43 ***

Response: (0,1,1)
  Item i Item j Item k Obs  Exp (O-E)^2/E
1      7      9     10  12 0.18    761.43 ***
2      6      9     10  15 0.47    449.34 ***
3      8      9     10  11 0.64    168.67 ***

Response: (1,1,1)
  Item i Item j Item k Obs   Exp (O-E)^2/E
1      3      9     10 106 62.62     30.05 ***
2      5      9     10  89 52.13     26.08 ***
3      2      9     10  99 60.74     24.10 ***

'***' denotes a chi-squared residual greater than 3.5
```

8.2.2.2 Plotting

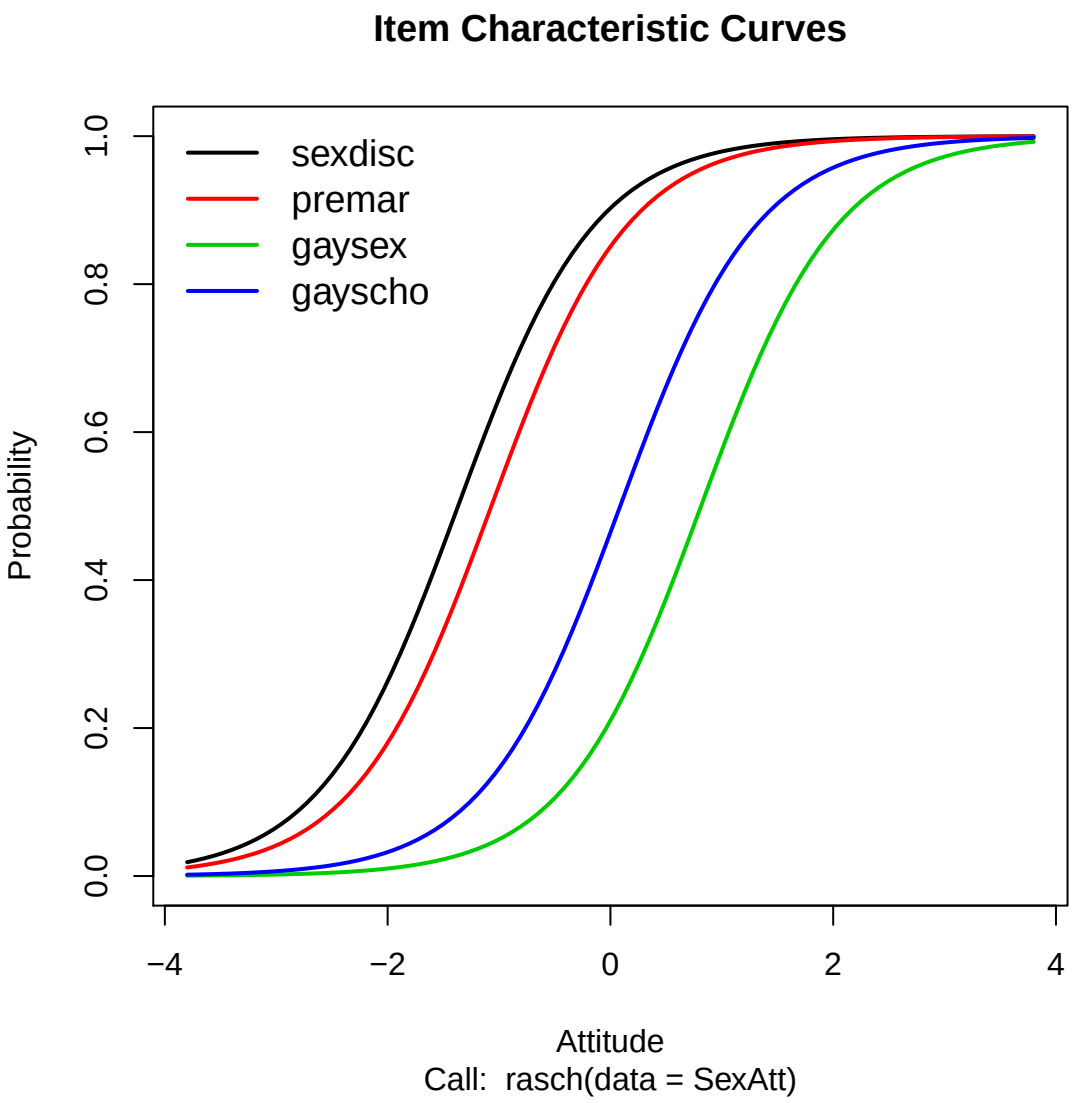
Even though the models we presented do not provide the optimal fit to the data at hand, for illustrative purposes we will produce several figures to depict the output of the fitted unconstrained Rasch model. The analogous plots for the other IRT models we fitted can be produced in exactly the same manner. The output of function `rasch` is an object of class 'rasch' for which the `plot()` method produces the Item Characteristics Curves that describe how the probability of a positive response relates to the levels of the latent sexual attitude.

```
R> plot(fit.rasch2, legend = TRUE, pch = rep(1:2, each = 5),
+      xlab = "Attitude", col = rep(1:5, 2), lwd = 2, cex = 1.2,
+      sub = paste("Call: ", deparse(fit.rasch2$call)))
```



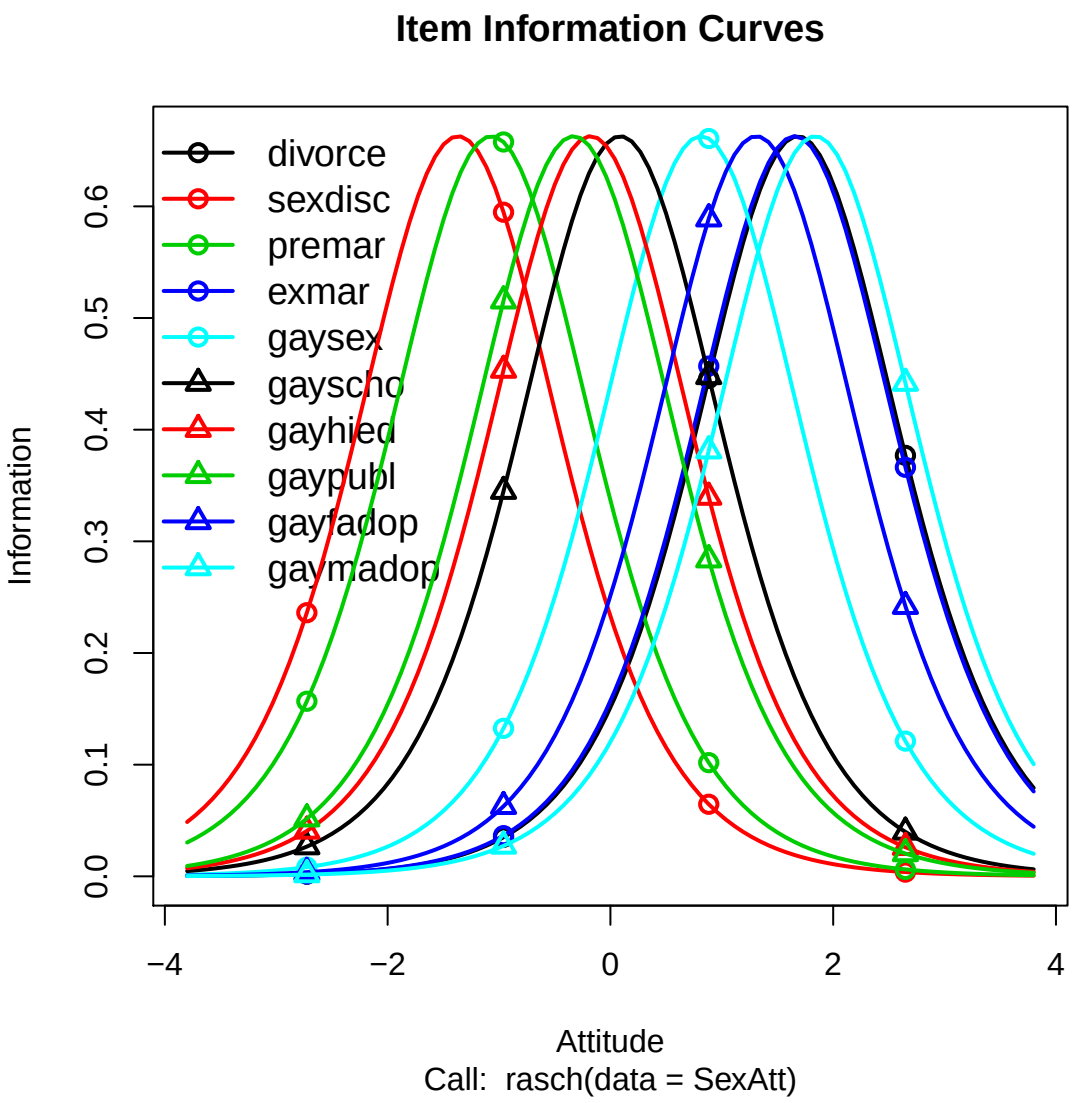
By default the item characteristic curves of all items are produced. In the case we analyze many items this would turn up to be a very busy plot. To overcome this the `plot()` method for fitted IRT models has the optional argument `items` that can be used to specify which items we would like to include in the plot. For example, a plot that only contains the item characteristic curves of items 2, 3, 5, and 6 is produced by

```
R> plot(fit.rasch2, legend = TRUE, items = c(2,3,5,6),
+      xlab = "Attitude", lwd = 2, cex = 1.2,
+      sub = paste("Call: ", deparse(fit.rasch2$call)))
```



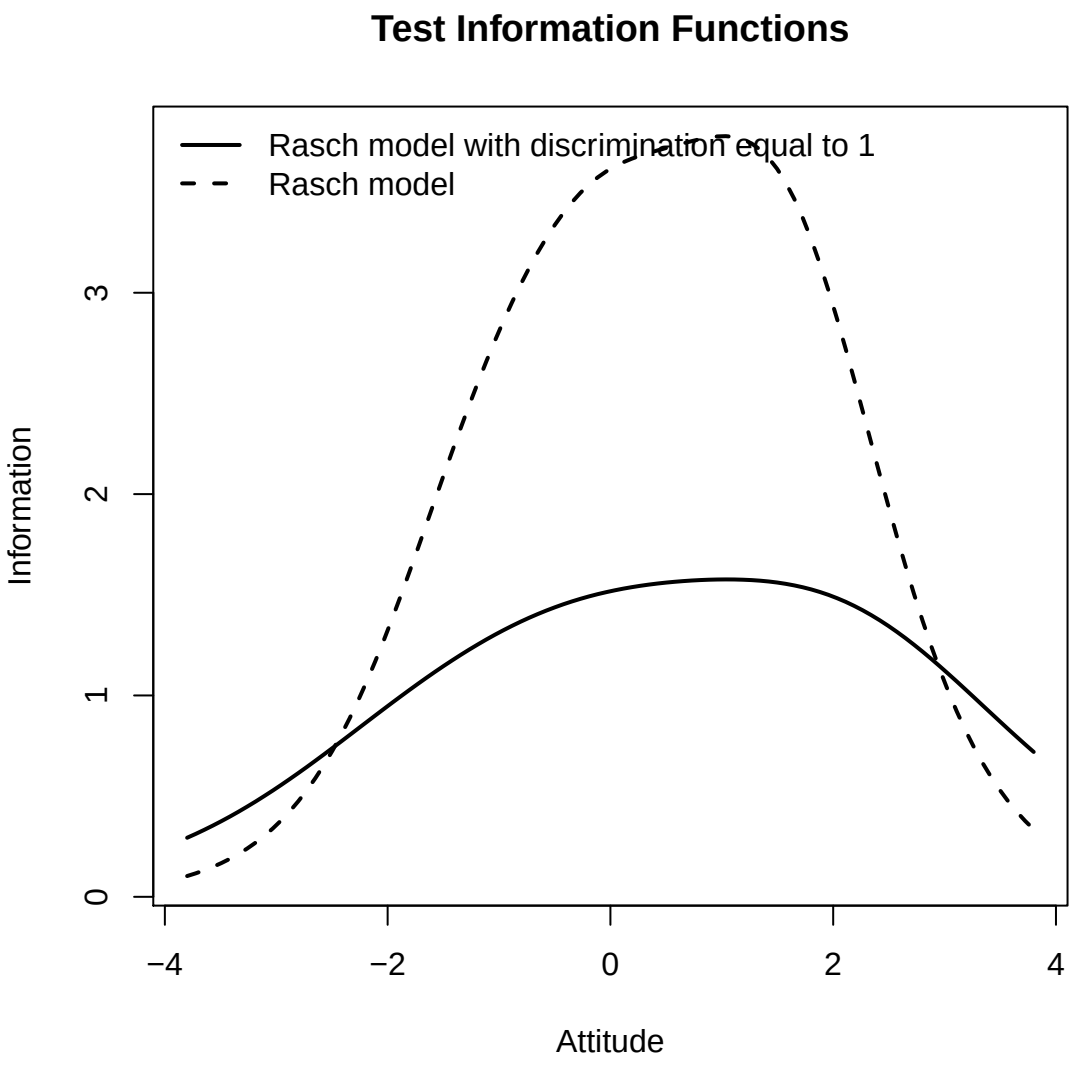
Following we depict the Item Information Curves, which are produced with almost the same syntax as the item characteristic curves. The only difference is that we need to change the default value of the `type` argument:

```
R> plot(fit.rasch2, type = "IIC", legend = TRUE, pch = rep(1:2, each = 5),
+      xlab = "Attitude", col = rep(1:5, 2), lwd = 2, cex = 1.2,
+      sub = paste("Call: ", deparse(fit.rasch2$call)))
```



For the Test Information Curve, which is the sum of the item information curves, we further need to specify the value 0 for the `items` argument. With the following syntax we in fact compare the test information curves of the constrained and unconstrained Rasch models:

```
R> info1 <- plot(fit.rasch1, type = "IIC", items = 0, lwd = 2, xlab = "Attitude",
+      plot = FALSE)
R> info2 <- plot(fit.rasch2, type = "IIC", items = 0, lwd = 2, xlab = "Attitude",
+      plot = FALSE)
R> # put them in the same plot
R> plot(range(info1[, "z"]), range(info1[, "info"], info2[, "info"]), type = "n",
+      xlab = "Attitude", ylab = "Information", main = "Test Information Functions")
R> lines(info1, lwd = 2)
R> lines(info2, lwd = 2, lty = 2)
R> legend("topleft", c("Rasch model with discrimination equal to 1", "Rasch model"),
+      lty = 1:2, lwd = 2, bty = "n")
```



In order to quantify the information provided by a set of items using a particular IRT model, `ltm` has the function `information()` that numerically approximates the area under the item or test information curves. For instance, we can compare the amount of information from items 2, 3 and 5 provided in the latent variable range (0, 4) from the constrained and unconstrained Rasch models:

```
R> information(fit.rasch1, range = c(0, 4), items = c(2, 3, 5))

Call:
rasch(data = SexAtt, constraint = cbind(ncol(SexAtt) + 1, 1))

Total Information = 3
Information in (0, 4) = 1.01 (33.57%)
Based on items 2, 3, 5

R> information(fit.rasch2, range = c(0, 4), items = c(2, 3, 5))

Call:
rasch(data = SexAtt)

Total Information = 4.89
Information in (0, 4) = 1.68 (34.35%)
Based on items 2, 3, 5
```

8.2.2.3 Estimating Latent Abilities

Estimates for the latent variable for each subject can be obtained using function `factor.scores()`. This function accepts as main argument a fitted IRT model. By default ability estimates are returned for all observed response patterns using the maximum aposteriori method as described in Section 8.1.4. For instance, for the unconstrained Rasch model we have

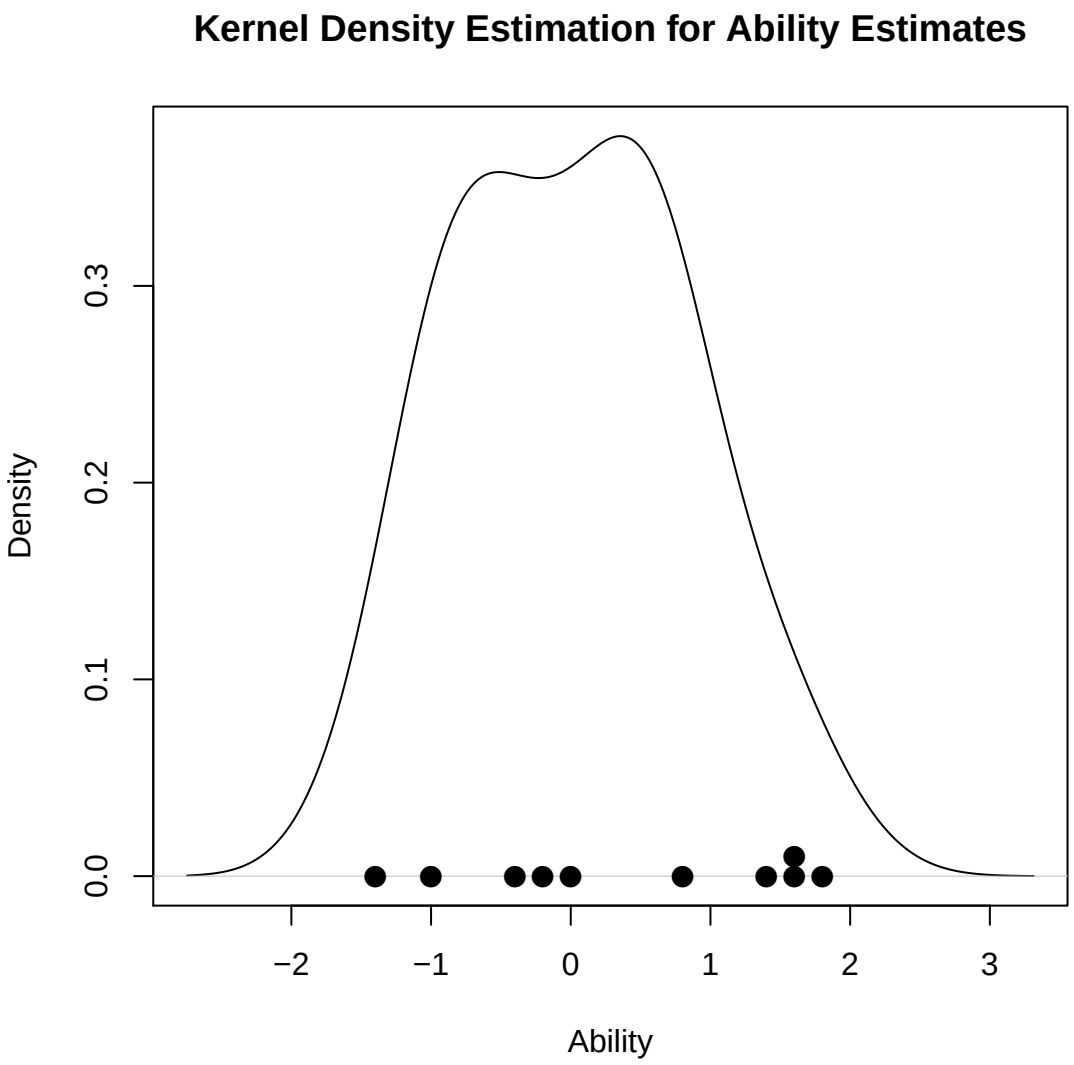
```
R> fsc <- factor.scores(fit.rasch2)
R> fsc

Call:
rasch(data = SexAtt)

Scoring Method: Empirical Bayes

Factor-Scores for specified response patterns:
  divorce sexdisc premar exmar gaysex gayscho gayhied gaypubl gayfadop gaymadop
1       0       0      0     0      0       0       0       0        0        0
2       0       0      0     0      0       0       0       0        1        0
3       0       0      0     0      0       0       0       0        1        1
  Obs    Exp     z1 se.z1
1  29 54.342 -1.613 0.586
2   1  0.629 -1.119 0.523
3   1  0.007 -0.702 0.491

...
```

If it is desired to produce the ability estimates for specific responses patterns (such as for the original data set), then this can be specified via the argument `resp.patterns`. In addition, the default maximum a posteriori method for estimating the latent abilities can changed using the `method` argument. For example, we produce the expected a posteriori estimates of the latent variable, for the patterns of all-positive and all-negative responses using the following syntax

```
R> factor.scores(fit.rasch2, resp.patterns = rbind(rep(1, 10), rep(0, 10)),
+      method = "EAP")

Call:
rasch(data = SexAtt)

Scoring Method: Expected A Posteriori

Factor-Scores for specified response patterns:
  divorce sexdisc premar exmar gaysex gayscho gayhied gaypubl gayfadop gaymadop
1       1       1      1     1      1       1       1       1        1        1
2       0       0      0     0      0       0       0       0        0        0
  Obs    Exp     z1 se.z1
1   5 11.192  2.260 0.553
2  29 54.342 -1.705 0.610
```

Furthermore, there is also a `plot()` method for objects returned by functions `factor.scores()`. This produces the kernel density estimation plot of the ability estimates, and if the optional argument `include.items` is set to TRUE it also superimposes the grouped item difficulty estimates (this is similar to the item-persons maps plot).

```
R> plot(fsc, include.items = TRUE)
```



8.3 IRT Analysis for Polytomous Data

The data we consider here come from the Environment section of the 1990 British Social Attitudes Survey (Brook et al., 1991; Bartholomew et al., 2002). The data frame `Environment` available in `ltm` contains the responses of 291 individuals asked about their opinion on six environmental issues. The response options were "very concerned", "slightly concernedänd "not very concerned"giving thus rise to six ordinal items.

8.3.1 Descriptive Analysis

As for the Sexual Attitude data, the `descript()` function can be used to produce descriptive statistics for the Environment data-set (output not shown). We can observe that for all six items the first response level has the highest frequency, followed by the second and third levels. The p-values for the pairwise associations indicate significant associations between all items. An alternative method to explore the degree of association between pairs of items is the computation of a nonparametric correlation coefficient. The `rcor.test()` function provides this option:

```
R> rcor.test(Environment, method = "kendall")
```

```
             LeadPetrol RiverSea RadioWaste AirPollution Chemicals Nuclear
LeadPetrol    *****      0.385    0.260      0.457        0.305     0.279
RiverSea     <0.001      *****    0.399      0.548        0.403     0.320
RadioWaste   <0.001     <0.001    *****      0.506        0.623     0.484
AirPollution <0.001     <0.001   <0.001      *****        0.504     0.382
Chemicals    <0.001     <0.001   <0.001     <0.001        *****     0.463
Nuclear      <0.001     <0.001   <0.001     <0.001       <0.001     *****

upper diagonal part contains correlation coefficient estimates
lower diagonal part contains corresponding p-values
```

The implementation of `rcor.test()` is based on the `cor()` function of package `stats` and thus it provides two options for nonparametric correlation coefficients, namely the Kendall's *tau* and the Spearman's *rho*, controlled by the `method` argument. The `print()` method for class 'rcor.test' returns a square matrix in which the upper diagonal part contains the estimates of the correlation coefficients, and the lower diagonal part contains the corresponding *p*-values.

8.3.2 IRT Models for Polytomous Data

8.3.2.1 Fitting

We start our analysis fitting Generalized Partial Credit Models. Function `gpcm()` can fit several versions of this model controlled by the value of the `method` argument:

```
R> fit.gpcm1 <- gpcm(Environment, constraint = "rasch")
R> fit.gpcm2 <- gpcm(Environment, constraint = "1PL")
R> fit.gpcm3 <- gpcm(Environment, constraint = "gpcm")
```

In particular, when `constraint = "rasch"`, then in the GPCM (8.4) we make the constraint $\alpha_1 = \ldots \alpha_p = 1$. When `constraint = "1PL"`, then in (8.4) we make the constraint that $\alpha_1 = \ldots \alpha_p = \alpha$. Finally, when `constraint = "gpcm"` (the default), then we impose no constraint in the values of $\alpha_1, \ldots, \alpha_p$. As it is the case for almost all model fitting functions in R, the `summary()` method returns more detailed output. For 'gpcm' objects the `summary()` method has the optional logical argument `robust.se`, which if set to `TRUE`, then standard errors are based on the sandwich estimator, e.g.,

```
R> summary(fit.gpcm3, robust.se = TRUE)
```

```
Call:
gpcm(data = Environment, constraint = "gpcm")

Model Summary:
   log.Lik      AIC      BIC
 -1094.974 2225.947 2292.067

Coefficients:
$LeadPetrol
        value std.err z.value
Catgr.1 0.613   0.163   3.770
Catgr.2 2.355   0.302   7.810
Dscrmn  1.168   0.228   5.127

$RiverSea
        value std.err z.value
Catgr.1 1.122   0.162   6.942
Catgr.2 2.373   0.286   8.284
Dscrmn  2.148   0.452   4.748

$RadioWaste
        value std.err z.value
```

```
Catgr.1 0.827   0.115   7.174
Catgr.2 1.694   0.158  10.695
Dscrmn  2.814   0.566   4.970

$AirPollution
        value std.err z.value
Catgr.1 0.466   0.088   5.309
Catgr.2 2.128   0.204  10.449
Dscrmn  3.100   0.651   4.760

$Chemicals
        value std.err z.value
Catgr.1 0.848   0.105   8.064
Catgr.2 1.738   0.164  10.618
Dscrmn  2.716   0.486   5.592

$Nuclear
        value std.err z.value
Catgr.1 0.230   0.133   1.726
Catgr.2 1.232   0.166   7.427
Dscrmn  1.441   0.240   6.004


Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 0.02
optimizer: nlminb
```

Due to the fact that `fit.gpcm1` is nested in `fit.gpcm2`, and `fit.gpcm2` is nested in `fit.gpcm3`, we can perform likelihood ratio tests in order to check if increasing the complexity of the model increases sufficiently the likelihood:

```
R> anova(fit.gpcm1, fit.gpcm2)

 Likelihood Ratio Table

              AIC     BIC  log.Lik   LRT df p.value
fit.gpcm1 2318.35 2362.43 -1147.18       12
fit.gpcm2 2248.03 2295.78 -1111.02 72.32 13  <0.001

R> anova(fit.gpcm2, fit.gpcm3)

 Likelihood Ratio Table

              AIC     BIC  log.Lik   LRT df p.value
fit.gpcm2 2248.03 2295.78 -1111.02       13
fit.gpcm3 2225.95 2292.07 -1094.97 32.08 18  <0.001
```

We observe that the unconstrained GPCM provides, represented by `fit.gpcm3` provide the best fit to the data at hand compared to the constrained versions. The overall fit of the model can be checked using functions `margins()` and `GoF.gpcm()`. As in `GoF.rasch()`, `GoF.gpcm()` uses the Pearson χ^2 goodness-of-fit measure but it provides both the Bootstrap (default) and the asymptotic approximations of the distribution of this statistic. This is controlled by the logical argument `simulate.p.value`. For instance, for `fit.gpcm3` we have

```
R> GoF.gpcm(fit.gpcm3, simulate.p.value = FALSE)

Pearson chi-squared Goodness-of-Fit Measure

Call:
gpcm(data = Environment, constraint = "gpcm")
```

```
Tobs: 777.4
df: 710
p-value: 0.04
```

whereas the `margins()` method produces:

```
R> margins(fit.gpcm3)

Call:
gpcm(data = Environment, constraint = "gpcm")

Fit on the Two-Way Margins

             LeadPetrol RiverSea RadioWaste AirPollution Chemicals Nuclear
LeadPetrol   -          14.09     4.14       9.17         4.09      3.55
RiverSea                -         9.27      10.04         6.00      5.65
RadioWaste                       -          5.93        12.54     11.41
AirPollution                               -            7.31      6.47
Chemicals                                                -         3.03
Nuclear                                                           -

R> margins(fit.gpcm3, type = "three-way")

Call:
gpcm(data = Environment, constraint = "gpcm")

Fit on the Three-Way Margins

   Item i Item j Item k (O-E)^2/E
1       1      2      3     33.29
2       1      2      4     31.51
3       1      2      5     40.20
4       1      2      6     29.69
5       1      3      4     28.59
6       1      3      5     39.42
7       1      3      6     30.14
8       1      4      5     30.73
9       1      4      6     27.51
10      1      5      6     15.57
11      2      3      4     35.00
12      2      3      5     30.93
13      2      3      6     32.54
14      2      4      5     28.79
15      2      4      6     27.26
16      2      5      6     20.20
17      3      4      5     42.40
18      3      4      6     36.71
19      3      5      6     33.09
20      4      5      6     33.31
```

We continue the analysis of the data set using the GRM (8.3). Function `grm()` provides the option to fit two versions of (8.3), namely the constrained GRM in which $\alpha_1 = \ldots \alpha_p = \alpha$, and the unconstrained GRM in which we assume a different discrimination parameter per item. These are fitted with the following syntax:

```
R> fit.grm1 <- grm(Environment, constrained = TRUE, Hessian = TRUE)
R> fit.grm2 <- grm(Environment, Hessian = TRUE)
R> summary(fit.grm2)

Call:
grm(data = Environment, Hessian = TRUE)

Model Summary:
   log.Lik      AIC      BIC
 -1090.404 2216.807 2282.927
```

```
Coefficients:
$LeadPetrol
        value std.err z.vals
Extrmt1 0.487   0.122  3.988
Extrmt2 2.584   2.699  0.957
Dscrmn  1.378   0.212  6.514

$RiverSea
        value std.err z.vals
Extrmt1 1.058   0.120  8.801
Extrmt2 2.499  23.516  0.106
Dscrmn  2.341   0.393  5.950

$RadioWaste
        value std.err z.vals
Extrmt1 0.779   0.092  8.446
Extrmt2 1.793  13.553  0.132
Dscrmn  3.123   0.516  6.053

$AirPollution
        value std.err z.vals
Extrmt1 0.457   0.085  5.399
Extrmt2 2.157  55.375  0.039
Dscrmn  3.283   0.575  5.710

$Chemicals
        value std.err z.vals
Extrmt1 0.809   0.094  8.592
Extrmt2 1.868  12.791  0.146
Dscrmn  2.947   0.485  6.077

$Nuclear
        value std.err z.vals
Extrmt1 0.073   0.101  0.716
Extrmt2 1.427   0.736  1.939
Dscrmn  1.761   0.248  7.102


Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 0.0030
quasi-Newton: BFGS
```

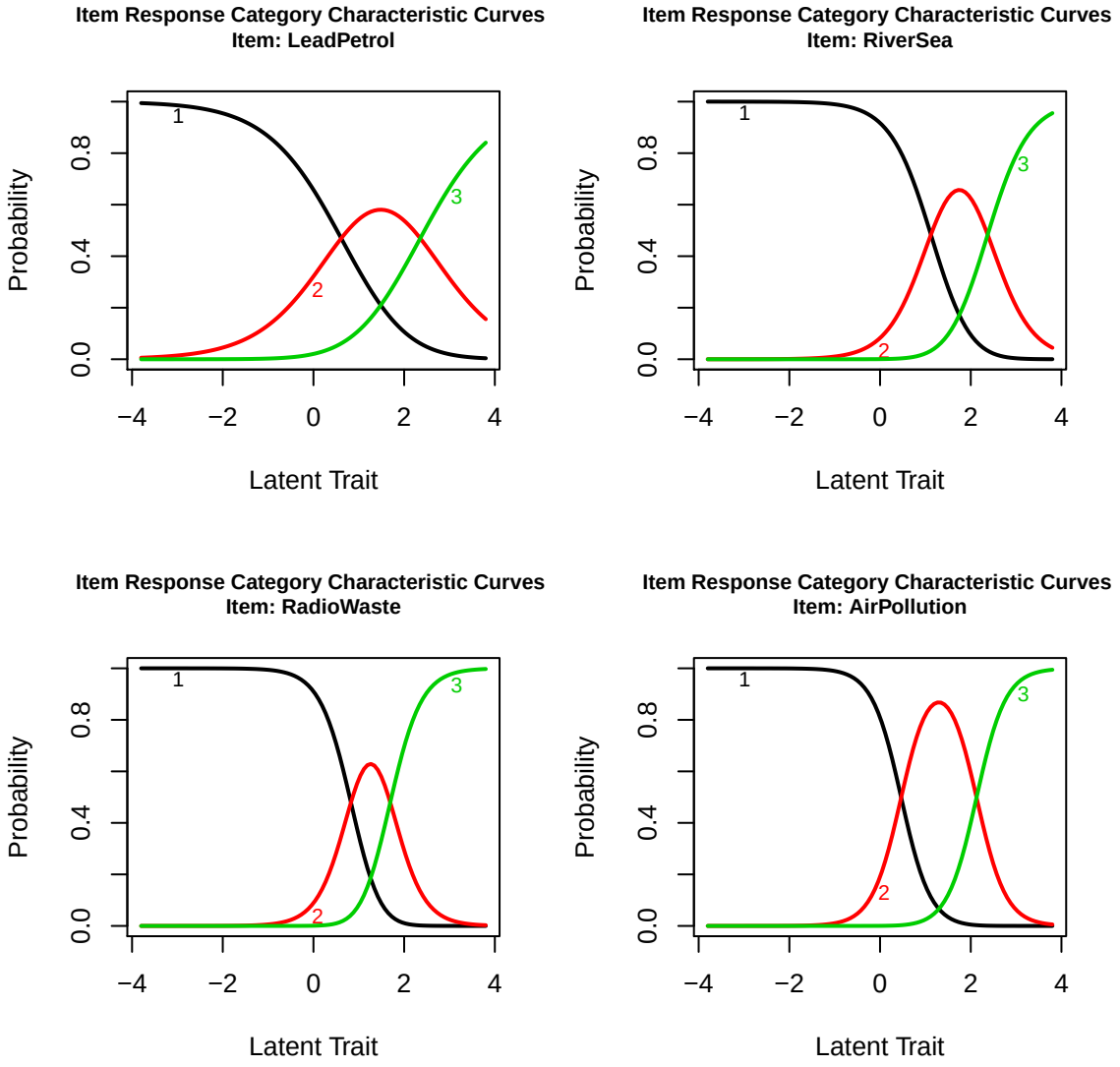
As with the GPCM, the `margins()` function can be used to test the fit of the graded response model in the two- and three-way margins. Moreover, method `anova()` performs a likelihood ration between the constrained and unconstrained models, e.g.,

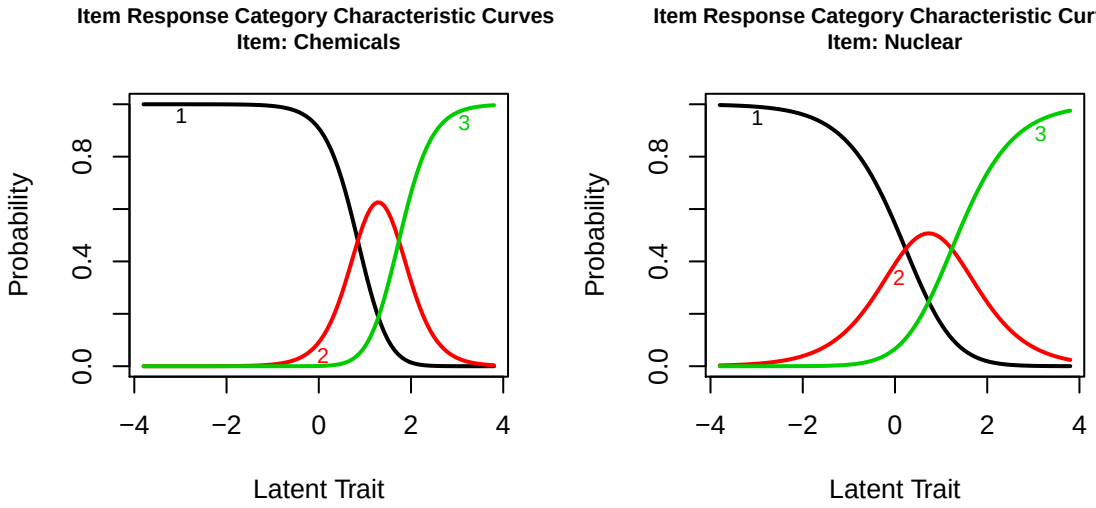
```
R> anova(fit.grm1, fit.grm2)

 Likelihood Ratio Table
             AIC     BIC  log.Lik   LRT df p.value
fit.grm1 2238.39 2286.14 -1106.19
fit.grm2 2216.81 2282.93 -1090.40 31.58  5  <0.001
```

8.3.2.2 Plotting

Plotting proceeds in the same manner as in the IRT models for dichotomous data. In particular, by default a call to the `plot()` method for objects of either class 'gpcm' or 'grm' will produce the item response category characteristic curves.

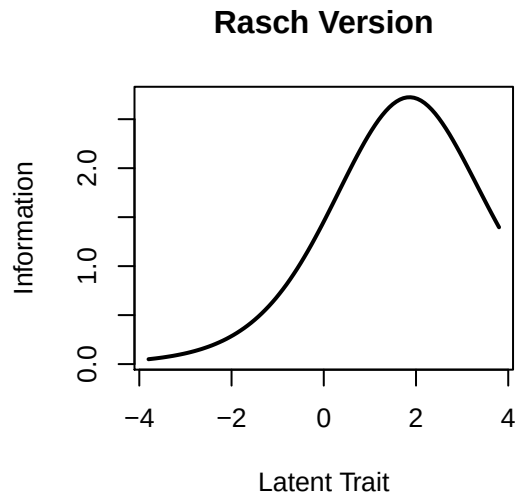
```
R> par(mfrow = c(2, 2))
R> plot(fit.gpcm3, lwd = 2, cex = 0.8, xlab = "Latent Trait")
```

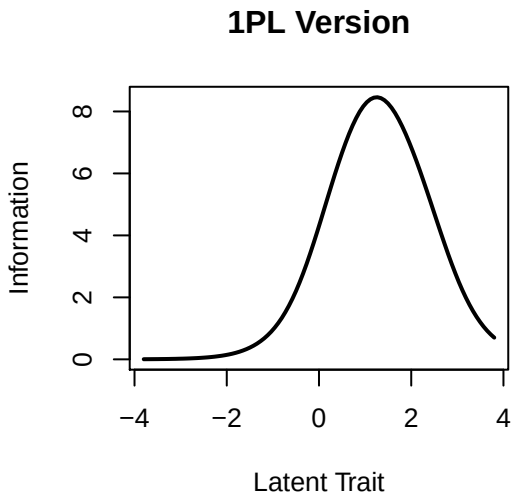


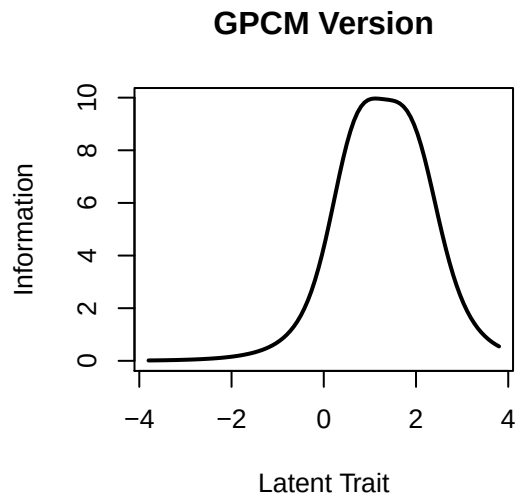


Similarly, the item information curves are produced using `plot(..., type = "IIC")`, whereas for the test information curve we also need to set argument `items` to zero. The following syntax compares graphically the test information functions of the three versions of the GPCM fitted above.

```
R> par(mfrow = c(2, 2))
R> plot(fit.gpcm1, type = "IIC", items = 0, lwd = 2, xlab = "Latent Trait",
+      main = "Rasch Version")
R> #
R> plot(fit.gpcm2, type = "IIC", items = 0, lwd = 2, xlab = "Latent Trait",
+      main = "1PL Version")
R> #
R> plot(fit.gpcm3, type = "IIC", items = 0, lwd = 2, xlab = "Latent Trait",
+      main = "GPCM Version")
```



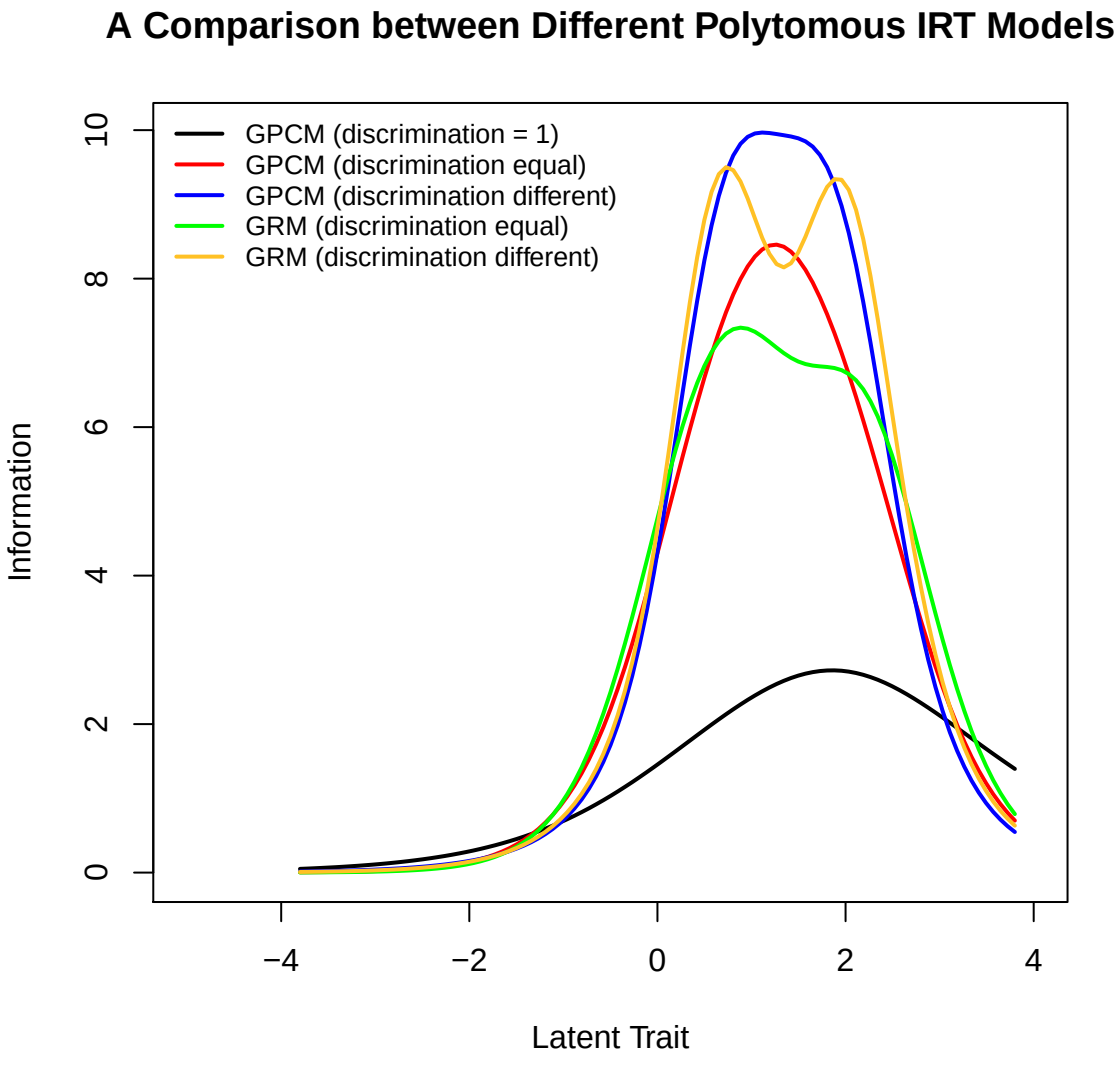




A more direct comparison is achieved by superimposing the test information function in the same plot. To do this we need a few extra steps (we also include the test information functions of the two versions of the graded response model), namely using the `plot()` but with the argument `plot` set to `FALSE`, we obtain a matrix with the values of the information function for a set of values for the latent ability. Following we put the information functions of all models in a matrix and we plot the result using `matplot()`:

```
R> vals1 <- plot(fit.gpcm1, type = "IIC", items = 0, plot = FALSE)
R> vals2 <- plot(fit.gpcm2, type = "IIC", items = 0, plot = FALSE)
R> vals3 <- plot(fit.gpcm3, type = "IIC", items = 0, plot = FALSE)
R> vals4 <- plot(fit.grm1, type = "IIC", items = 0, plot = FALSE)
R> vals5 <- plot(fit.grm2, type = "IIC", items = 0, plot = FALSE)
R> values <- cbind(vals1[, "test.info"], vals2[, "test.info"],
+     vals3[, "test.info"], vals4[, "test.info"], vals5[, "test.info"])
R> matplot(vals1[, "z"], values, type = "l",
+     col = c("black", "red", "blue", "green", "goldenrod1"),
+     xlab = "Latent Trait", ylab = "Information",
+     main = "A Comparison between Different Polytomous IRT Models",
+     lty = 1, lwd = 2)
R> lgd <- c("GPCM (discrimination = 1)", "GPCM (discrimination equal among items)",
+     "GPCM (discrimination different)",
+     "GRM (discrimination equal)",
+     "GRM (discrimination different)")
```

```
R> legend("topleft", lgd, col = c("black", "red", "blue", "green",
+     "goldenrod1"), cex = 0.8, lty = 1, lwd = 2, bty = "n")
```



8.3.2.3 Estimating Latent Abilities

As with plotting fitter IRT models for polytomous data, estimation of latent abilities proceeds in the same manner as for dichotomous data. Namely, abilities are estimated using function `factor.scores()`, in which available options are expected a posteriori and maximum a posteriori. Below we present a comparison of the empirical Bayes estimates for the latent for the five polytomous IRT models fitted to the Environment data set. We estimate abilities for three subjects one who has chosen option "very concerned" for all items, one who has chosen option "slightly concerned" for all items, and one who has chosen option "not very concerned" for all items.

```
R> resp.patts <- rbind(rep(1, 6), rep(2, 6), rep(3, 6))
R> models <- list(fit.gpcm1, fit.gpcm2, fit.gpcm3, fit.grm1, fit.grm2)
R> out <- sapply(models, function (x) {
+     factor.scores(x, resp.patts)$score.dat$z1
+ })
R> dimnames(out) <- list(paste("All", 1:3), c("GPCM(rasch)", "GPCM(1PL)",
+     "GPCM", "GRM(constr)", "GRM"))
R> out

      GPCM(rasch)  GPCM(1PL)      GPCM GRM(constr)        GRM
All 1   -0.861761 -0.7729611 -0.723661  -0.7604717 -0.7259074
All 2    1.366235  1.1554543  1.191593   1.1760591  1.2080896
All 3    3.060339  2.6642328  2.611972   2.7577941  2.6792566
```

Kapitel 9
RUMM – Rasch Unidimensional Measurement Models

Thomas Salzberger

9.1 Grundlagen

RUMM ist ein Akronym und steht für *Rasch Unidimensional Measurement Models*. Die Software wurde von David Andrich, Barry Sheridan und Guanzhong Luo entwickelt. Die erste veröffentlichte Version (RUMM 2.7q) datiert auf 1997 zurück. In der Folge wurden die Versionen 2010, 2020 und – Anfang des Jahres 2010 – die Version RUMM 2030 (Andrich et al., 2010) vom RUMM Laboratory Pty Ltd (http://www.rummlab.com.au/) publiziert.

Ein wesentliches Merkmal von RUMM 2030 ist seine Benutzerfreundlichkeit. Der Datenimport, die Auswahl der Analysen, sowie die Abfrage der Ergebnisse erfolgt durchgängig menügesteuert, erfordert also keine programmspezifische Syntaxformulierung. Das Windows-Programm liefert gut gestaltete, in der Regel druckreife grafische Outputs, die auch leicht exportiert und in Textverarbeitungsprogramme problemlos eingebunden werden können.

Im Unterschied zu SPSS, existieren keine RUMM-Syntaxdateien und grundsätzlich auch keine Outputfiles. Ergebnisse lassen sich jedoch gegebenenfalls in textbasierte Dateien oder Grafikfiles ausgeben. Jeder Datensatz stellt ein eigenes *Projekt* (project) dar. RUMM speichert alle zu einem Projekt gehörenden Information (Daten, Analysen) in einem Projektfile mit der Endung *mbd*.

9.2 Modelle

RUMM schätzt ausschließlich (eindimensionale) Rasch-Modelle, also Item Response Theory Modelle, die der Parameterinvarianz bzw. der spezifischen Objektivität (Rasch, 1977) entsprechen. Das einfachste Modell ist daher das Rasch-Modell für dichotome Daten (Rasch, 1980). Daneben können polytome Rasch-Modelle (Andrich, 1978b,a, 1988a; Masters, 1982) geschätzt werden, wobei unterschiedliche Einschränkungen im Bezug auf die Thresholdparameter möglich sind (beispielsweise gleiche Thresholdabstände über die Items hinweg, wie dies im sogenannten Ratingscale-Modell der Fall ist). Die Zahl der Kategorien kann bis zu 101 Antwortoptionen betragen, sodass auch Visual Analogue Scales (VAS, Wewers und Lowe, 1990) analysiert werden können. Neben mehrkategoriellen Items können auch Multiple Choice-Formate verarbeitet werden. Die verschiedenen Itemarten bzw. Antwortformate können zudem leicht in einer Analyse kombiniert werden. Schließlich können auch Multifacet-Designs gerechnet werden. Selbstverständlich können in einem Projekt auch Items zusammengefasst werden, die verschiedenen Dimensionen zuzurechnen sind. Im Rahmen jeder Analyse können die entsprechenden Items ausgewählt werden. Alle Analysen werden im Projektfile gespeichert und sind jederzeit wieder abrufbar.

thomas.salzberger@wu.ac.at

Log-lineare Testmodelle (LLTM) sind in RUMM allerdings nicht implementiert. Generell lassen sich IRT-Modelle mit unterschiedlicher Itemdiskriminanz (allgemeine IRT-Modelle, OPLM) nicht schätzen, ebensowenig nichtkumulative Modelle (unfolding models, Andrich, 1989, 1996). Das vor kurzem vorgestellte Frame-of-Reference Rasch model (Humphry, 2005) ist in der aktuellen Version von RUMM noch nicht implementiert.

9.3 Schätzmethode

RUMM wendet die paarweise Schätzmethode an (Zwindermann, 1995; Andrich und Luo, 2003), welche ebenso wie das Conditional Maximum Likelihood-Verfahren eine Separierbarkeit der Parameter gewährleistet und keinerlei Verteilungsannahmen trifft.

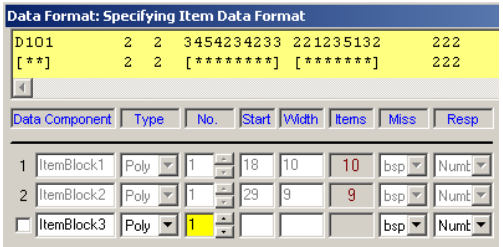
9.4 Dateninput

Der Dateninput erfolgt in Form von „fixed format ASCII files", welche innerhalb von RUMM definiert werden müssen. SPSS-Dateien, Microsoft Excel-Dokumente oder Datenformate anderer Programme können nicht direkt eingelesen werden. In diesem Fall ist zunächst die Ausgabe der Daten in ein ASCII-File erforderlich. Obwohl dies zunächst als eine nachteilige Barriere erscheinen mag, so garantiert die bewusste Auswahl von Variablen im Allgemeinen eine sinnvolle Auswahl von Variablen, die auch tatsächlich analysiert werden können.

Die Definition von Datenfiles in RUMM ist sehr übersichtlich gestaltet. Zunächst ist die Struktur der Daten festzulegen, wobei hier lediglich anzugeben ist, ob eine Personen-ID vorhanden ist oder nicht bzw. ob neben der Item-Facette weitere Variables definiert werden sollen. Letzteres ist bei multi-facet-Modellen der Fall. So kann beispielsweise neben der Personenfähigkeit (oder – allgemeiner formuliert – der Ausprägung der zu messenden latenten Variablen bei den Personen) und der Itemschwierigkeit (Ausprägung der Items auf der latenten Variablen) auch eine Raterentität (wie sie etwa bei der Beurteilung von Essays von Bedeutung ist) definiert werden.

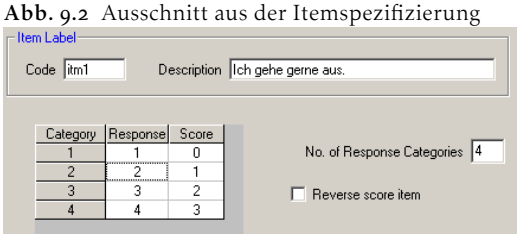
Im nächsten Schritt ist die Personen-ID anzugeben (wenn zuvor festgelegt), sowie weitere die Personen charakterisierenden Faktoren (typischerweise soziodemografische Merkmale oder Designvariablen bei experimentellen Settings). Schließlich ist anzugeben, welche Positionen im File die eigentlichen Responsedaten darstellen. Da RUMM eine Zeile des Datenfiles exemplarisch anzeigt, können die entsprechenden Positionen sehr einfach mit der Maus markiert werden. Es müssen keine numerischen Positionsangaben gemacht werden. Abbildung 9.1 zeigt einen Ausschnitt aus der Datendefinition. „D101" wurde als Personen-ID festgelegt, zwei Datenblöcke stellen zehn bzw. neun Items umfassende Blöcke dar. All diese Angaben können in einem Template-File abgespeichert werden, sodass bei einer neuerlichen Definition des Datenfiles das Template-File herangezogen werden kann und das Projekt so schnell neu erstellt werden kann.

Abb. 9.1 Ausschnitt aus dem Datendefinitionsfenster



Die Definition der einzelnen Items (Label; Typ: kategoriell oder multiple choice; Anzahl der Kategorien und deren Scoring; umgekehrt gepolt oder nicht) erfolgt in einem separaten Schritt. Abbildung 9.2 zeigt die Spezifizierung des Items „itm1" mit der Bezeichnung „Ich gehe gerne aus.". Die vierstufige Antwortskala wurde ursprünglich mit 1 bis 4 kodiert, wird in RUMM nun zu 0 bis 3 umkodiert. Auch hier kann eine Template-Datei abgespeichert werden. Template-Files sind auch direkt mit jeden Editor oder auch MS Excel editierbar.

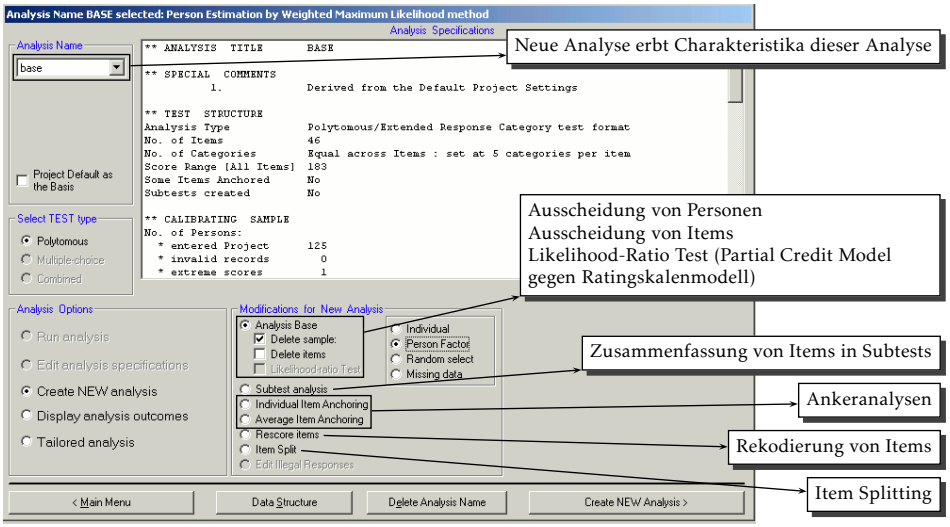
Abb. 9.2 Ausschnitt aus der Itemspezifizierung



9.5 Analysen

Jede einzelne Modellschätzung stellt eine *Analyse* (analysis) dar. Grundsätzlich werden alle Personen und alle im Projekt enthaltenen Items in der Weise herangezogen, wie sie definiert wurden. Jede neue Analyse kann jedoch in vielerlei Hinsicht modifiziert werden (siehe Abbildung 9.3).

Abb. 9.3 Erstellung einer neuen Analyse



Erstens kann die Basis der Analyse geändert werden. Einerseits kann die Stichprobe der Personen eingeschränkt werden. Hier können individuelle Respondenten ausgeschieden werden oder Einschränkungen aufgrund von Personenfaktoren vorgenommen werden (z.B. nur Frauen, oder nur Angehörige einer bestimmten Nationalität, etc.), vorausgesetzt diese wurden bei der Projekterstellung entsprechend definiert. Schließlich ist auch eine zufällige Auswahl von Datensätzen möglich. Andererseits können individuelle Items ausgewählt werden bzw. bestimmte Items aus der Analyse ausgeschieden werden. Natürlich können Modifikationen der Analysebasis im Hinblick auf Personen und Items auch gleichzeitig vorgenommen werden.

Zweitens kann eine Subtest-Analyse verlangt werden. Dabei werden Items zu Superitems (oft auch als Testlets oder Item-Bundles bezeichnet) zusammengefasst. Beispielsweise kann so aus zwei dichotomen Items mit den Scores 0 und 1, ein kombiniertes Item mit den Scores 0, 1 und 2 generiert werden. Solche Analysen können beispielsweise dann erforderlich sein, wenn Items der lokalen stochastischen Unabhängigkeit nicht entsprechen. Auch im Rahmen der Untersuchung der Eindimensionalität können Subtest-Analysen aufschlussreiche Ergebnisse liefern.

Drittens sind Ankeranalysen (anchoring) möglich, wobei RUMM zwischen Individual Item Anchoring und Average Item Anchoring unterscheidet. Bei diesen Analysen werden Parameterwerte für (einzelne oder alle) Items vorgegeben und der Fit auf der Grundlage dieser Parameterwerte bestimmt. Ankerwerte können auch relativ spezifiziert werden, also Unterschiede zwischen zwei Items betreffen.

Viertens kann das Scoring der Items im Rahmen einer Rescoring-Analyse geändert werden. Dabei wird der Scoringschlüssel nachträglich adaptiert. Abgesehen von Fehlkodierungen (z.B. falsche Kodierung eines umgekehrt gepolten Items) sind vor allem Verletzungen der Ordnung der Thresholdparameter bei multikategoriellen Items Anlass für solche Analysen. Stellt sich nämlich heraus, dass die Personen zwischen zwei Responsekategorien nicht differenzieren, so ist eine unterschiedliche Scorevergabe nicht sinnvoll. RUMM ermöglicht es daher, mehrer Kategorien gleich zu scoren.

Fünftens können einzelne Items gesplittet werden. Dazu ist die Angabe eines Personenfaktors erforderlich, sowie dessen Ausprägungen, für die ein Item geteilt werden soll. Item Splits sind dann angezeigt, wenn ein Item in Teilstichproben einen unterschiedlichen Lageparameter aufweist (Differential Item Functioning, DIF). Uniformes DIF kann so sehr leicht berücksichtigt werden und ein Item kann mit Hilfe gruppenspezifischer Parameter trotzdem beibehalten werden. Freilich sollte sich der Nutzer darüber im Klaren sein, dass dies streng genommen eine Verletzung der spezifischen Objektivität darstellt. Eine plausible (oder besser noch theoretisch fundierte) Erklärung für DIF sollte daher stets vorliegen.

Nachdem die entsprechende Analyse ausgewählt wurde (und durch einen Bezeichnung eindeutig festgelegt wurde), können weitere Analysespezifikationen geändert werden. Dies betrifft zunächst die Wahl des Modells für polytome Daten (unrestringiertes Partial Credit Model („unrestricted") oder Ratingskalenmodell („rating")). Auch die Struktur der Thresholds lässt sich bestimmen. So kann hier beispielsweise eine gleiche Varianz der Thresholdparameter über alle Items festgelegt werden. Die anderen Einstellungen betreffen die Konvergenzkriterien und die Schätzung der Personenparameter, wobei bei letzteren zwischen „Weighted Likelihood" und „Maximum Likelihood" gewählt werden kann.

Nach dem Rechnen der Analyse wird durch „Display Analysis" zum Ergebnisfenster gewechselt. Zuvor können noch Einstellungen zu den Fittests vorgenommen werden.

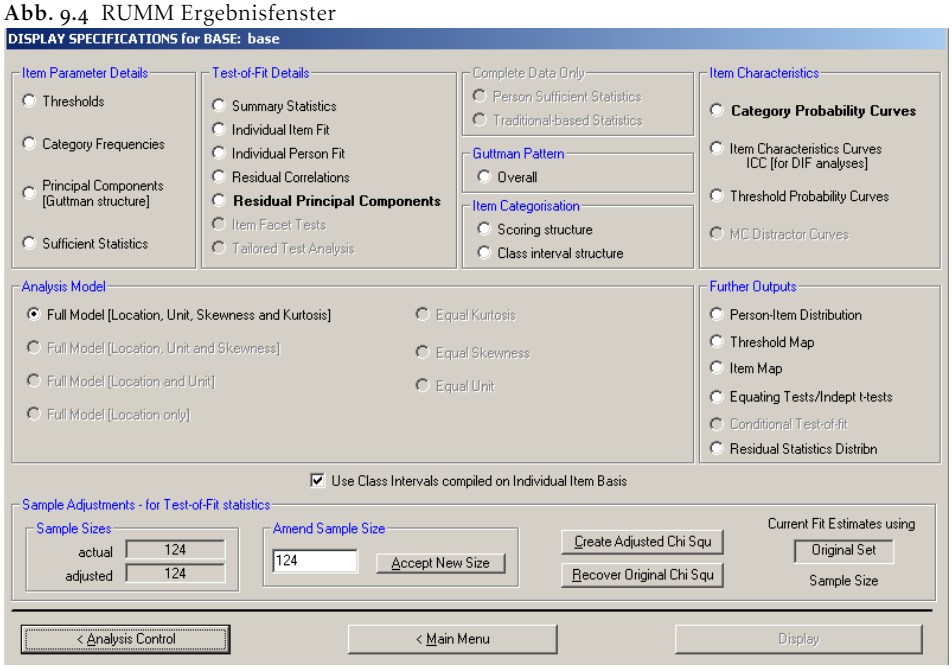
Das Ergebnisfenster (siehe Abbildung 9.4) listet alle Ergebnisoptionen auf. Die gewünschten Ausgaben können durch entsprechendes Anklicken angefordert werden.

9.6 Ergebnisse

9.6.1 Item Parameter Details

Dieser Block bietet Ausgaben zu den Itemparametern (Threshold-Parameter bei mehrkategoriellen Items), den Häufigkeiten in den einzelnen Responsekategorien, sowie den Hauptkomponenten und den suffizienten Statistiken der Thresholdmodellierung in RUMM. RUMM schätzt die einzelnen

Abb. 9.4 RUMM Ergebnisfenster



Thresholdparameter nicht direkt, sondern modelliert die Verteilungsmomente der Thresholds pro Item.

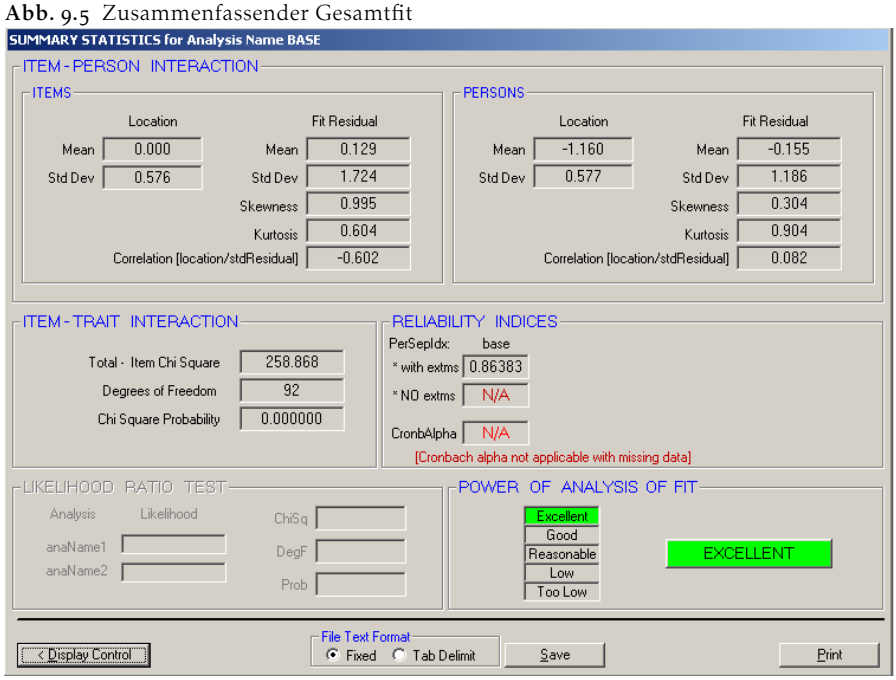
9.6.2 Fittests

Eine zentrale Ausgabeeinheit betrifft die Fittests. Die Summary Fit Statistics liefern eine Übersicht über die Fitstatistiken über alle Items bzw. über alle Personen, sowie betreffend die Interaktion der Items und der Personen. Bei letzterem handelt es sich um ein totales Chi-Quadrat, welches die Fitstatitiken der einzelnen Items aggregiert. Bei dieser Gesamtfitstatistik ist dementsprechend ein nicht-signifikantes Chi-Quadrat wünschenswert.

Für die Items wird neben dem Mittelwert der Itemparameter (welcher mit Ausnahme von Ankeranalysen stets 0 ist) auch deren Standardabweichung angegeben. Aus dieser wird ersichtlich, ob die Items auch einen entsprechend großen Bereich der latenten Variable abdecken. Das Item-Fitresiduum quantifiziert die tatsächliche Diskriminanz des Items. Ein Wert von 0 entspricht der unter dem Raschmodell erwarteten Diskriminanz (die Steigung der Itemcharakteristikkurve, ICC, entspricht dem Raschmodell). Positive Werte stehen für Unterdiskriminanz (zu flache ICC), negative Werte zeigen Überdiskriminanz an (zu steile ICC). Über alle Items wird eine Standardabweichung der Fitresiduen von 1 erwartet. Deutlich höhere Werte implizieren, dass einige Items über- und andere unterdiskriminieren. Dies kann, muss aber nicht eine Folge von Mehrdimensionalität sein.

Für die Personenparameter werden die entsprechenden Statistiken ebenfalls angegeben. Ein Personenparametermittelwert, der sich deutlich vom Itemparametermittelwert (der in der Regel 0 ist) unterscheidet, weist auf ein mangelndes Targeting hin (schlechte Übereinstimmung von Item- und Personenparameter). Die Fitresiduen der Personen geben Aufschluss darüber, wie sehr die Antwortmuster der einzelnen Respondenten mit dem Modell übereinstimmen. Negative Werte stehen für eine zu starke Annäherung an den deterministischen Guttman-Pattern der Antworten. Positive Werte implizieren Personenmisfit im Sinne eines zu stark zufallsbedingten Antwortmusters. Theoretisch sollte auch der Mittelwert der Personenfitresiduen 0 sein. In der Praxis weist RUMM jedoch oft leicht negative Werte aus (in der Regel um −0.2 oder −0.3). Dementsprechend sind Mittelwerte von 0 bereits

Abb. 9.5 Zusammenfassender Gesamtfit



ein Alarmsignal im Sinne von Personenmisfit.

Schließlich werden im Bereich Reliabilitätsindizes der Person-Separation-Index (PSI, Andrich, 1982) angeführt, welcher im Sinne der Definition der Reliabilität, die Truescore-Varianz im Verhältnis zur totalen Varianz angibt. Cronbach's alpha wird nur bei vollständigen Datensätzen ausgewiesen.

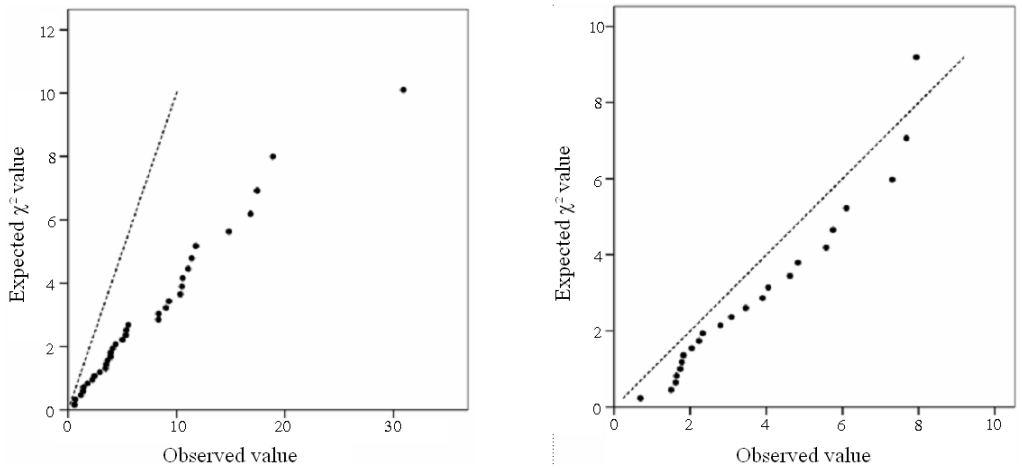
Die Angaben zur Power der Analyse beruhen auf dem PSI, wobei bei Werten von mindestens 0.85 eine exzellente Power ausgewiesen wird. Die Person-Separation (also die Reliabilität im Sinne des Raschmodells) beruht im Wesentlichen darauf, wie gut das Messinstrument zwischen den Personen differenziert. Bei geringer Personenvarianz, aber auch bei ungünstigem Targeting ist der PSI entsprechend geringer. In der Folge sind die Fittests weniger mächtig. Ein geringer PSI weist also auf Probleme in den Daten hin, die allerdings nicht sehr spezifisch sind und von Mistargeting und geringer Personenvarianz über übermäßige Fehlervarianz bis hin zu völlig ungeeigneten Items, die keine gemeinsame Dimension abbilden, reichen können.

Der individuelle Itemfit wird tabellarisch ausgegeben. Für jedes Item wird ein Lageparameter ausgewiesen (bei polytomen Antwortformaten handelt es sich dabei um den Mittelwert der Thresholdparameter), sowie ein Standardschätzfehler. Schätzfehler für einzelne Thresholdparameter (welche im Bereich Item Parameter Details angezeigt werden) sucht man in RUMM allerdings vergeblich. Dem statistischen Itemfit (im Unterschied zur weiter unten angeführten grafischen Modellkontrolle) wird durch drei Größen Rechnung getragen. Das Itemfitresidual quantifiziert die Steigung der ICC und damit die empirische Diskriminanz des Items. Diese Statistik ist standardisiert auf einen erwarteten Wert von 0 bei perfektem Fit und näherungsweise normalverteilt. Vor allem positive Werte über 2.5 weisen auf Items hin, die zu schwach diskriminieren. Werte von kleiner als –2.5 zeigen Überdiskriminanz an. Solche Items sind zwar ebenfalls nicht modellkonform, werden aber im Zweifel (etwa wenn wenig Items zur Verfügung stehen) oft beibehalten.

Eine Chi-Quadrat-Statistik beruht auf einem Vergleich erwarteter und tatsächlicher mittlerer Responses in Personengruppen mit ähnlichen Personenparametern, welche zu Klassenintervallen zusammengefasst werden. RUMM ermöglicht auch einen Einblick in die Berechnung der Statistik, sodass die Beiträge der einzelnen Gruppen genau analysiert werden können. Signifikante Werte weisen auf Misfit der Items hin. Die Addition aller Item-Chi-Quadrate führt zur Gesamtfitstatistik,

die bei den „Summary Statistics" angegeben ist. Allerdings ist es durchaus möglich, dass sämtliche Items p-Werte von über 0.05 aufweisen, der Gesamtfit aber dennoch ungenügend ist. Diese Situation tritt dann auf, wenn sich die p-Werte der einzelnen Items nicht erwartungskonform über das gesamte Spektrum erstrecken, sondern allesamt nur knapp über dem Kriterium von 0.05 liegen. Es empfiehlt sich jedenfalls die Erstellung eines Q-Q-Plots der Chi-Quadratwerte, wie er beispielsweise in SPSS leicht generierbar ist (siehe Abbildung 9.6). Die Chi-Quadratwerte der einzelnen Items können aus RUMM mit „copy&paste" sehr leicht exportiert werden.

Abb. 9.6 In SPSS erstellte Q-Q-Plots vor (links) und nach (rechts) einer Bereinigung des Itempools



Schließlich ist eine *F*-Test-Statistik für jedes Item erhältlich. Sie muss allerdings vor der Anzeige aller Ergebnisse explizit verlangt werden. Diese Statistik berücksichtigt die einzelnen Personenparameter besser als die Chi-Quadrat-Statistik, welche alle Personen innerhalb einer Klasse als gleich ansieht. Bei idealem Fit ist der Mittelwert des Residuums in allen Klassenintervallen gleich 0 und damit in allen Gruppen gleich. Bei Misfit treten Gruppenunterschiede auf, die zu einem signifikanten Haupteffekt des Klassenintervalls führen. Die *F*-Test-Statistik lässt sich allerdings nicht über alle Items aggregieren.

Der individuelle Personfit liefert Fitinformationen für jede einzelne Person, sofern es sich nicht um einen extremen Antwortvektor handelt.

Die Residualkorrelationen der Items stellen ein wichtiges Prüfkriterium für lokale stochastische Unabhängigkeit der Items dar. Bei völliger Unabhängigkeit sollten diese Korrelationen 0 sein. Tatsächlich ist dies allerdings auch bei perfektem Fit nicht möglich, da die Residuen zweier Items durch die Bedingung eines konstanten Gesamtscores stets negativ korrelieren. Dies ist vor allem bei einer geringen Zahl an Items problematisch. Im Unterschied zur Interpretation von Korrelationen im Allgemeinen, sind Residualkorrelationen von mehr als 0.2 bis 0.3 bereits als sehr hoch anzusehen. Aufgrund der erwähnten Asymmetrie sind positive Korrelationen schon ab 0.2 kritisch, während negative Werte bis –0.3 tolerierbar sind. Hinweisen auf lokale stochastische Abhängigkeit sollte jedenfalls nachgegangen werden. Wenn es sich tatsächlich lediglich um eine logische Responseabhängigkeit handelt (eine positive Antwort bei einem Item impliziert eine positive Antwort bei einem anderen Item), so bietet RUMM durch eine Subtest-Analyse ein geeignetes Mittel zur Berücksichtigung an.

Korrelierende Residuen können allerdings auch auf Multidimensionalität hinweisen, welche freilich auch bei unauffälligen Residualkorrelationen ein Problem darstellen kann. In RUMM basiert die Überprüfung der Eindimensionalität im Wesentlichen auf einer Hauptkomponentenanalyse der Residuen (Smith, 2002). Dabei sollten sich keinerlei systematische Muster zeigen, die Eigenwerte der Komponenten sollten den zufällig zu erwartenden Werten entsprechen (welche am besten durch eine Parallel-Analyse (Allan und Hubbard, 1986; Watkins, 2000) ermittelt werden). Ein übermäßig hoher

erster Eigenwert ist meist ein Zeichen von Multidimensionalität. In diesem Fall sollten die unrotierten Ladungen der einzelnen Items auf der ersten Komponente besonders beachtet werden. Cluster von Items mit hoch-positiven bzw. hoch-negativen Ladungen bilden jeweils verschiedene Dimensionen ab. Die Klärung der Frage, inwieweit sich Multidimensionalität tatsächlich auswirkt, erfordert allerdings weitergehende Analyseschritte. Eine Möglichkeit stellt wiederum die Subtest-Analyse dar. Werden alle Items einer Dimension zu einem Subtest zusammengefasst und werden diese (in Regel) zwei Subtests reanalysiert, so wird die Reliabilität gegenüber der ursprünglichen Analyse stark absinken, da die beiden Dimensionen im ungünstigsten Fall gar nicht mehr korrelieren werden.

Eine weitere Möglichkeit, den Effekt der Multidimensionalität abzuschätzen, bietet RUMM im Bereich Equating an. Diese Analyse dient grundsätzlich dafür, Subsets von Items auszuwählen und die Entsprechung von Itemrohscores und metrischer Maße zu erhalten, wie dies beim Testequating erforderlich ist. Werden die beiden Itemsets allerdings so gewählt, dass eine Gruppe von Items lediglich eine Dimension erfasst (der Input dazu kann aus der Hauptkomponentenanalyse gewonnen werden) und die andere Gruppe die zweite Dimension, so lassen sich pro Person zwei Parameter schätzen. Bei perfekter Eindimensionalität dürfen sich diese Schätzwerte nur zufällig unterscheiden (Smith, 2002). RUMM berechnet pro Person eine t-Statistik (formal ein t-Test für unabhängige Stichproben), die den Unterschied der Parameter auf Signifikanz prüft. Dabei werden die Standardschätzfehler für die Berechnung des Standardfehlers der Differenz herangezogen. Es wird nun erwartet, dass nur in 5% (bzw. 1%) der Fälle der p-Wert kleiner oder gleich 5% (bzw. 1%) ist. Diese Erwartung wird den tatsächlichen Unterschieden gegenübergestellt. Auch für die Personenparameterschätzungen insgesamt wird ein t-Wert (abhängiger t-Test) ausgewiesen.

Die auf dem t-Test basierende Methode kann selbstverständlich auch theoriegeleitet angewendet werden. In diesem Fall erfolgt die Zusammenfassung der Items theoriegestützt statt datengesteuert. Letztlich hilft diese Methode die praktische Auswirkung von (möglicher) Multidimensonalität abzuschätzen. Als alleiniges Verfahren zur Eindimensionalitätsprüfung greift dieser Ansatz wohl etwas zu kurz, zumal er insbesondere bei höher korrelierten Dimensionen rasch an Mächtigkeit verliert.

9.7 Grafische Ausgaben

Ein weiterer zentraler Ausgabebereich in RUMM ist der Block Itemcharakteristiken. In diesem Block sind grafische Ausgaben abrufbar. Itemcharakteristkurven (siehe Abbildung 9.7) stellen den Erwartungswert des manifesten Scores in Abhängigkeit vom Personenparameter dar. Für zu Klassen zusammengefasste Personen mit ähnlichem Parameter werden die tatsächlichen Itemscores (empirische Itemcharakteristiken) angegeben. Aus dem Vergleich dieser Scores mit den erwarteten Werten ist ein grafischer Modelltest möglich. Die Zahl der Klassenintervalle wird von RUMM so gewählt, dass in jeder Klasse rund 60 Personen zusammengefasst werden. Diese Vorgabe kann allerdings vom Benutzer auch geändert werden.

Abb. 9.7 ICC eines Items mit tatsächlichen mittleren Scores in drei Personengruppen

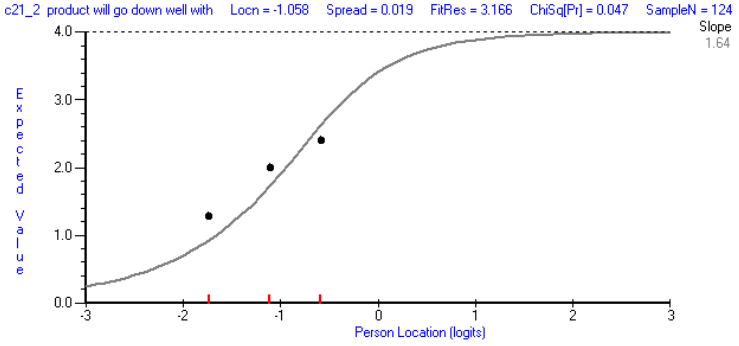
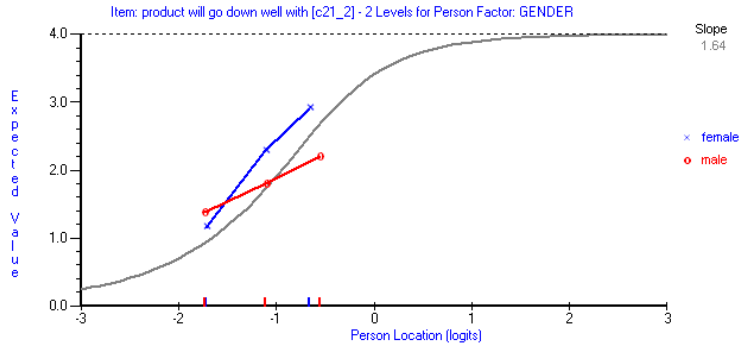


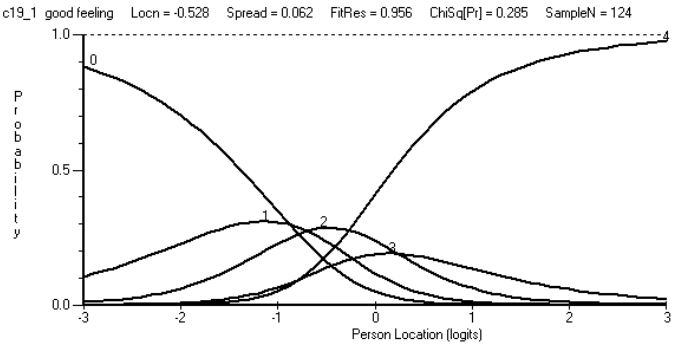
Abb. 9.8 ICC eines Items und empirische Verläufe für Männer und Frauen



Aufschlussreich ist die Darstellung von tatsächlichen Scores in Abhängigkeit von Personenfaktoren. So können beispielsweise empirische Itemcharakteristikkurven für Männer und Frauen angezeigt werden oder für verschiedene Nationalitäten. Voraussetzung ist lediglich, dass diese Variablen als Personenfaktoren definiert wurden. Durch eine zweifache Varianzanalyse (neben dem Klassenintervall ist der Personenfaktor ein Faktor in der Varianzanalyse) können Unterschiede zwischen Respondentengruppen im Sinne des Differential Item Functioning (DIF) auf Signifikanz geprüft werden. Ein signifkanter Haupteffekt für den Personenfaktor spricht für uniformes DIF, eine signifikante Wechselwirkung für nicht-uniformes DIF. Letzteres bedeutet eine unterschiedliche Steigung der ICC innerhalb der Personengruppen und impliziert damit einen qualitativen Bedeutungsunterschied des Items, welcher auch durch ein Item-Splitting nicht berücksichtigt werden kann. Uniformem DIF hingegen kann in der Regel durch eine Item-Splitting-Analyse entsprochen werden, wenngleich dies, wie bereits erwähnt, ein Abgehen von strikter spezifischer Objektivität bedeutet.

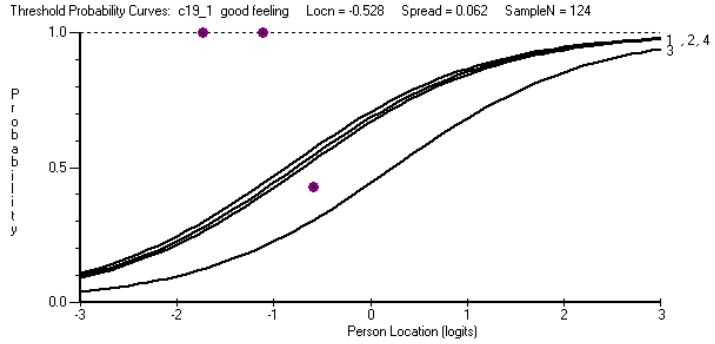
Besonders attraktiv ist die Möglichkeit, die DIF-Analysen mit einem Knopfdruck für alle Items durchzuführen. Die Ergebnisse werden in Tabellenform dargestellt, signifikante Effekte lassen sich – mit oder ohne Bonferroni-Korrektur des Signifikantzniveaus - farblich hervorheben.

Abb. 9.9 Kategoriecharakteristikkurven



Kategoriecharakteristikkurven geben Einblick in die Funktionsweise aller einzelnen Antwortkategorien (siehe Abbildung 9.9). Ungeordnete Thresholdparameter sind hier leicht grafisch erkennbar. Eine Antwortkategorie wird in diesem Fall nie zur wahrscheinlichsten Antwortoption (siehe Kategorie 3 in Abbildung 9.9). Numerisch ist dies im Bereich Item Parameter Details / Thresholds tabellarisch dargestellt, wobei ungeordnete Thresholds farblich automatisch hervorgehoben werden. Auch eine farbige Threshold Map, welche die Übergänge der einzelnen Antwortoptionen grafisch darstellt, lässt ungeordnete Thresholds sehr leicht erkennen. Ungeordnete Thresholds sollten nicht ignoriert werden (Andrich, 1995a,b), auch wenn sie nicht unbedingt Misfit implizieren (aber in den meisten Fällen doch zu unzureichendem Fit führen).

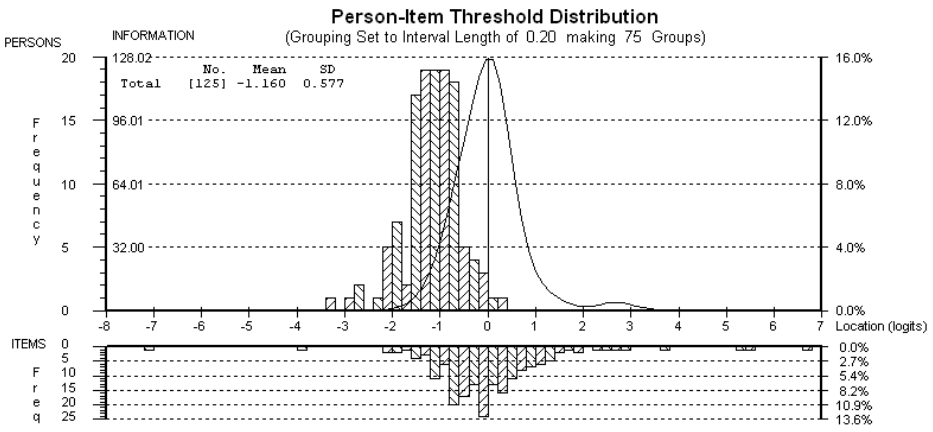
Abb. 9.10 Threshold Probability Kurven mit tatsächlichen Antworthäufigkeiten für die Threshold zwischen Kategorie 3 und 4



Der wichtigste Grund für eine falsche Ordnung der Thresholds liegt in der unzureichenden Diskriminanz einer Grenze zwischen zwei aufeinander folgenden Antwortkategorien. RUMM bietet diesbezüglich eine grafische Darstellung der Threshold Probability Kurven (siehe Abbildung 10) an. Für jede Kategoriengrenze wird eine ICC (wie im dichotomen Fall) angezeigt. Zusätzlich lassen sich die tatsächlichen Antworthäufigkeiten der Personen, die eine dieser beiden Kategorien gewählt haben, darstellen. So lassen sich sehr leicht dysfunktionale Kategorien identifizieren. In Abbildung 10 diskriminiert die Threshold zwischen den Kategorien 3 und 4 nicht. Durch nachträgliches Rescoring im Rahmen einer neuen Analyse lassen sich ungeordnete Thresholds in der Regel beheben, wenn gleich das geänderte Scoringformat durch neue Daten überprüft werden sollte. Ungeordnete Thresholds treten fast zwangsläufig auf, wenn zu viele Kategorien angeboten werden. Die Personen können in diesem Fall zwischen den angebotenen Responseoptionen nicht mehr adäquat differenzieren. Ein weiterer Grund sind unscharfe verbale Kennzeichnungen. So unterscheiden sich beispielsweise die Kategorien „ab und zu" und „gelegentlich" nicht wirklich, sodass es auch nicht gerechtfertigt ist, eine Kategorie höher zu scoren als die andere. Auch negativ formulierte Fragen führen überdurchschnittlich häufig zu ungeordneten Thresholds.

Einen guten Überblick über die Verteilung der Items, der Personen oder von Items und Personen bieten Targeting-Plots (siehe Abbildung 9.11). Hier können auch Teilausschnitte (z.B. nur Männer) oder auch Gegenüberstellungen (z.B. Männer versus Frauen) abgebildet werden. Mittelwertsunterschiede der Personengruppen lassen sich ebenfalls anzeigen und auf Signifikanz prüfen. Schließlich kann auch eine Informationsfunktion gezeichnet werden, welche eindrücklich die Leistungsfähigkeit des Messinstruments in Abhängigkeit von der Lage auf der latenten Dimension darstellt.

Abb. 9.11 Targeting Plot in RUMM



In jedem Ergebnisbereich können Tabellen oder Grafiken abgespeichert werden. Meist ist durch einfaches Kopieren und Einfügen auch ein Übertrag in Tabellenkalkulationsprogramme oder SPSS völlig unkompliziert möglich. Umfangreiche Datenausgaben einschließlich von Schätzwerten für Personen und deren Standardfehler sind über den Personenfit-Bildschirm möglich. So können Rasch-Personenparameter beispielsweise dem ursprünglichen SPSS-Datenfile hinzugefügt werden.

9.8 Resumee

RUMM 2030 ist ein außerordentlich benutzerfreundliches, leistungsfähiges Windowsprogramm zur Schätzung von einer Reihe unterschiedlicher Modelle aus der Rasch-Familie. Vor allem die grafischen Darstellungen erlauben einen sehr guten Einblick in die Funktionalität des Messinstruments. Die Bestimmung des Modellfits erfolgt auf der Basis von ausreichend vielen Fit-Statistiken, wobei die Grafiken stets mitberücksichtigt werden sollten. Umfangreich sind weiters die Möglichkeiten, die Datenbasis zu verändern (Bildung von Subtests, Item-Splitting, Rescoring, etc.).

Als Nachteil mag die fehlende Schnittstelle zu anderen Datenanalyseprogrammen wahrgenommen werden. In der Praxis bedeutet dies jedoch nur eine geringfügige Einschränkung, die den Nutzer dazu zwingt, genau zu überlegen, welche Daten eingelesen bzw. ausgegeben werden sollen.

Unbefriedigend ist das Fehlen von Standardschätzfehlern für die Thresholdparameter. Es ist zu hoffen, dass diese in zukünftigen Versionen angeboten werden wird. Allerdings bieten auch andere „kommerzielle" Rasch-Analyseprogramme dies, im Unterschied zum R-Package eRm, nicht an. Was die Prüfung der Eindimensionalitätsanforderung betrifft, so stehen andere explorative wie konfirmatorische Verfahren zur Verfügung, wie sie beispielsweise in DIGRAM implementiert sind.

Ein weiterer, allerdings nicht besonders störender Nachteil hängt mit der Zusammenfassung von allen Analysen in einem Projektfile zusammen. Zu Beginn sind Projektfiles sehr kompakt, da sie im Wesentlichen nur aus einem ASCI-Datenfile und den entsprechenden Definitionen bestehen. In der Folge wird jedoch offenbar für jede Analyse ein eigenes Datensetup dem Projekt hinzugefügt. Dies beschleunigt zwar die Ausgabe der Ergebnisse und ermöglicht eine schnelle Spezifikation einer weitere Analyse (die die Änderungen jeder beliebigen anderen Analyse „erben" kann), führt aber andererseits zu einem steten Anwachsen des Projektfiles. Auch das Löschen von Analysen ändert daran nichts.

Ein in der Praxis bedeutsameres Problem, vor allem bei methodischen Forschungsfragen, ist mit dem Fehlen einer editierbaren Syntax verbunden. Für den gewöhnlichen Nutzer ist es unzweifelhaft von Vorteil, sich nicht mit einer spezifischen Programmiersprache auseinander setzen zu müssen. Für versiertere Verwender impliziert dies freilich auch Grenzen. So ist es nicht möglich, eine Vielzahl von simulierten Datenfiles automatisiert einer Analyse in RUMM zuzuführen. Template-Files ermöglichen zwar eine rasche Definition formal identischer Projekte, großzahlige Simulationsstudien sind so aber nur sehr bedingt durchführbar.

Die Grundfunktionen von RUMM 2030 lassen sich relativ einfach mit Hilfe der online verfügbaren Manuale bedienen und nutzen. Fortgeschrittene Analysen lassen sich am besten durch einschlägige Workshops erlernen. Diesbezügliche Angebote finden sich auf der (nicht gewinnorientierten) Website http://www.matildabayclub.net. Der Matilda Bay Club ist ein interdisziplinärer Zusammenschluss von Forschern, die zur Messung latenter Variablen das Raschmodell nutzen und sich für dessen Diffusion in den Sozialwissenschaften einsetzen.

Kapitel 10
GGUM2004 – Generalized Graded Unfolding Models

Axel Sonntag und Daniela Weber

Zusammenfassung In diesem Kapitel wollen wir einen Einblick in das Programm GGUM2004 geben, das zur Analyse von Generalized Graded Unfolding Models (Entfaltungstechnik Modellen) von James S. Roberts, Haw-ren Fang, Weiwei Cui und Yingji Wang entwickelt wurde. Bevor wir jedoch näher auf das Programm eingehen wollen und dies auch anhand eines praktischen Beispiels testen werden, geben wir noch basierend auf Roberts et al. (2000) einen theoretischen Einblick in die vielleicht eher weniger bekannten Unfolding Modelle. In weiterer Folge wird die Benutzung des Programmes anhand des empirischen Beispiels erläutert, bevor abschließend alternative Softwarepakete kurz vorgestellt werden.

10.1 Die Methode der Enfaltungstechnik

In der empirischen Sozialforschung wurden auf Grund der verschiedensten Anwendungsbereiche unterschiedliche Messmodelle entwickelt. Ein sehr weit verbreitetes Verfahren ist die Guttman-Skala. Bei einer Guttman-Skala sind alle ICCs stufenförmig, da es hierbei nur eine richtige oder eine falsche Antwort gibt. Den Befragten werden Aussagen vorgelegt, denen sie zustimmen, oder die sie ablehnen dürfen. Da die Aussagen von der einfachsten bis hin zur schwierigsten geordnet sein müssen, kann man so leicht einen Vergleich zwischen Personen auf Basis des Personenscores herstellen. Aufgrund der Konstruktion (Ordinalskalenniveau) ist es jedoch nicht möglich einen näheren Einblick zu geben, das heißt um wie viel eine Person zum Beispiel mehr weiß als eine andere (Rost, 2004; Bühner, 2006).

Coombs adaptierte 1950 das Prinzip der Guttman-Skala und entwickelte die nach ihm benannte Coombs-Skala (Entfaltungstechnik). Im Unterschied zur Guttman-Skala, bei der nur zwischen stimme zu und stimme nicht zu unterschieden wurde, können nun Präferenzordnungen behandelt werden. Man geht davon aus, dass jeder Befragte einen Platz auf einem Einstellungskontinuum hat, der als Idealpunkt bezeichnet wird. Eine Präferenzordnung der einzelnen Items bzw. Aussagen entspricht hier somit nichts anderem als einer in diesem Idealpunkt gefalteten Skala, welche als I-Skala bezeichnet wird. Entfaltet entspricht dies einer Skala (J-Skala), welche die Items und aber auch den Idealpunkt beinhaltet. Man kann so aufgrund der Präferenzordnung, die der Befragte erstellt, auf dessen Position auf dem zu messenden Einstellungskontinuum rückschließen (Pepels, 2007).
Multidimensionale unfolding Modelle finden in der Multidimensionalen Skalierung bei Präferenz- und Entscheidungsdaten Anwendung, während sich aber unfolding Modelle in der empirischen Sozialforschung weniger durchsetzen konnten (de Leeuw, 2005).

Hier wollen wir eine Weiterentwicklung von Graded Unfolding Models (GUMs) (Roberts, 1995; Roberts und Laughlin, 1996) vorstellen. Generalized Graded Unfolding Models (GGUMs) wurden von Roberts et al. entwickelt und in Roberts et al. (2000) gemeinsam mit einer adäquaten Softwarelösung vorgestellt.

WU Wien, Institut für Arbeitsmarkttheorie und -politik • axel.sonntag@wu.ac.at
WU Wien, Institut für Statistik und Mathematik • daniela.weber@wu.ac.at

10.1.1 Generalized Graded Unfolding Models

Dieser speziellen Art von unfolding Modellen liegen vier Prämissen bezüglich des Antwortprozesses zu Grunde:

- Personen, die einem Item zustimmen, liegen auf dem latenten Kontinuum nahe bei einem Item, währenddessen Personen, die ein Item eher ablehnen, sich von diesem weiter entfernt befinden. Man kann also auf Grund der Differenz zwischen der Position des i-ten Items (δ_i) und der Position der j-ten Person (θ_j) auf den Grad der Zustimmung der Person zum Item schließen. (Analog zu der in Abschnitt 10.1 beschreiben Idealpunkt Theorie.)
- Jede beobachtbare Antwort entspricht zwei subjektiven Antworten. Je nach Lage auf dem Kontinuum (positive oder negative Differenz zwischen δ_i und θ_j) kann man zwischen Zustimmung von oben beziehungsweise Zustimmung von unten unterscheiden. Zur Veranschaulichung soll hier Abbildung 10.1 dienen, wobei die Abszisse die Differenz zwischen Personen- und Itemparameter und die Ordinate die Wahrscheinlichkeiten für eine subjektive Antwortkategorie darstellen. Die vertikalen Linien bezeichnen hierbei die Thresholds τ_{ik} und die Kategoriebeschriftung bezeichnen die am höchsten wahrscheinliche Antwortkategorie)
- Roberts et al. (2000) nehmen aufgrund der allgemeinen Gültigkeit zur Beschreibung der subjektiven Antworten das von Muraki (1992) definierte verallgemeinerte partial credit Model (GPCM).
- Die Thresholds τ_{ik}s sind symmetrisch um den Punkt $0 = \delta_i - \theta_j$.

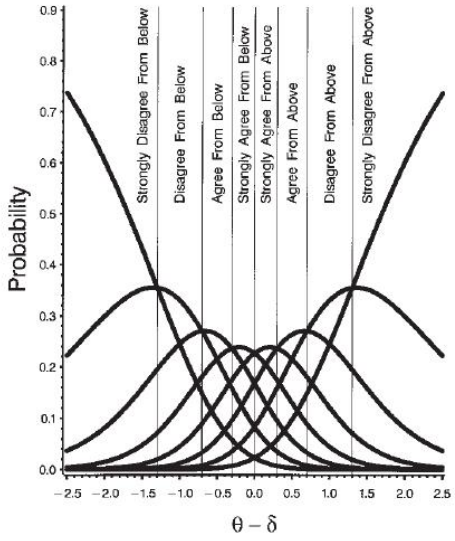


Abb. 10.1 Wahrscheinlichkeitsfunktionen der subjektiven Antwortkategorien. Je nach positiver oder negativer Differenz zwischen Personen- und Itemparameter, dargestellt auf der Abszisse, handelt es sich um eine Zustimmung bzw. Ablehnung von oben bzw. von unten. Auf der Ordinate befinden sich die Wahrscheinlichkeiten für eine subjektive Antwortkategorie. (aus Roberts et al. (2000), Seite 5, Figure 1)

Verwendet man nun das GPCM $P(Y_i = y|\theta_j) = \frac{\exp\{\alpha_i[y(\theta_j-\delta_i)-\sum_{k=0}^{y}\tau_{ik}]\}}{\sum_{w=0}^{M}\exp\{\alpha_i[w(\theta_j-\delta_i)-\sum_{k=0}^{w}\tau_{ik}]\}}$ und berücksichtigt, dass eine Modellierung für die beobachtbaren Antwortkategorien und nicht für die subjektiven Antwortkategorien stattfinden soll, so erhält man unter Einbeziehung der vier Prämissen das von Roberts et al. (2000) definierte GGUM:

$$P(Z_i = z|\theta_j) = \frac{\exp\{\alpha_i[z(\theta_j - \delta_i) - \sum_{k=0}^{z}\tau_{ik}]\} + \exp\{\alpha_i[(M-z)(\theta_j - \delta_i) - \sum_{k=0}^{z}\tau_{ik}]\}}{\sum_{w=0}^{C}\exp\{\alpha_i[w(\theta_j - \delta_i) - \sum_{k=0}^{w}\tau_{ik}]\} + \exp\{\alpha_i[(M-w)(\theta_j - \delta_i) - \sum_{k=0}^{w}\tau_{ik}]\}} \quad (10.1)$$

wobei Z_i die beobachtbare Antwort zum Einstellungsstatement i repräsentiert; $z = 0 \ldots C$ bezeichnet die Stärke der Nichtzustimmung, wobei 0 der stärkste Level der Nichtzustimmung ist; C ist die

Anzahl der beobachtbaren Antwortkategorien minus 1; α_i ist das Unterscheidungsvermögen des Einstellungsstatements i (der Diskriminationsparameter) und τ_{ik} beschreibt die Positionierung des $k$ten subjektiven Antwortkategorien Thresholds; M ist definiert als $M = 2C + 1$.

GGUM ist daher ein unfolding Model, bei dem sich die Wahrscheinlichkeit für die Wahl einer bestimmten Antwortkategorie von einer Person über die Summe der Wahrscheinlichkeiten für die beiden zugehörigen subjektiven Antwortkategorien berechnen lässt, wie man in Abbildung 10.2 erkennen kann. Es existieren daher nur halb so viele beobachtbare Antwortkategorien als subjektive.

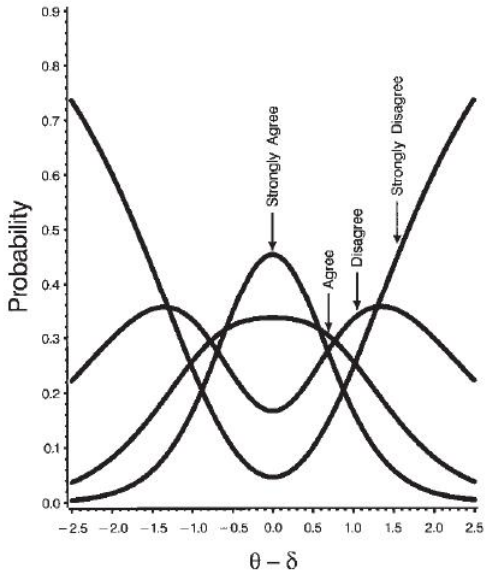


Abb. 10.2 Wahrscheinlichkeitsfunktionen der beobachtbaren Antwortkategorien, welche als die Summe der Wahrscheinlichkeiten der subjektiven Antwortkategorien von oben und unten, wie in Abbildung 10.1 dargestellt, berechnet werden können. (aus Roberts et al. (2000), Seite 7, Figure 2)

Diese Modelle wurden von Roberts et al. auch mit einem Interface in Fortran Code implementiert. Bevor wir auf das Programm näher eingehen wollen, widmen wir uns zunächst noch der Parameter-Schätzung.

10.1.2 Parameter-Schätzung

Die Schätzung der Item Parameter erfolgt mittels marginal maximum likelihood (MML) Methode, während die Personen Parameter mit expected a posteriori (EAP) Methode geschätzt werden.

10.1.2.1 Marginal Maximum Likelihood

Die Item Parameter werden mittels MML Methode geschätzt wobei der EM (expectation-maximization) Algorithmus Verwendung findet. Die Randwahrscheinlichkeit eine bestimmte Antwort einer Person, die gemäß einer kontinuierlichen Verteilung $g(\theta)$ einer Population entstammt, zu beobachten, entspricht $P(X_s) = \int_{-\infty}^{\infty} P(X_s|\theta)g(\theta)d\theta$.
Sei nun r_s die Anzahl an Personen mit dem Antwortvektor X_s und N die Anzahl an Personen, dann sei r_s multinominal verteilt. Die Likelihood sei dann
$L = \frac{N!}{\prod_{s=1}^{S} r_s!} \prod_{s=1}^{S} P(X_s)^{r_s}$ und daraus resultierend ergibt sich als Log-Likelihood:

$$\ln(L) = \ln(N!) - \sum_{s=1}^{S} \ln(r_s!) + \sum_{s=1}^{S} r_s \ln(P(X_s)). \tag{10.2}$$

Mit Hilfe der ersten partiellen Ableitung erhält man die MML Schätzer für die Item Parameter α_i, δ_i und τ_{ik}. Durch Anwendung des EM Algorithmus können im ersten Schritt, dem E-Schritt (estimation), die Schätzer für $\overline{r}_{izf}$ berechnet werden ($\overline{r}_{izf}$ bezeichnet hier die erwartete Häufigkeit für die Antwort z bei Item i mit Quadraturpunkt V_f). Der M-Schritt (maximization) wird anschließend solange durchgeführt bis die am besten geeigneten Item Parameter für alle Items bei gegebenen $\overline{r}_{izf}$ bestimmt werden konnten.

10.1.2.2 Expected a Posteriori

Nach der Berechnung der Item Parameter können mit Hilfe der beobachteten Antworten und einer expected a posteriori (EAP) Methode die Schätzer für die j-te Person

$$\hat{\theta}_j = \frac{\sum_{f=1}^{F} V_f L_j(V_f) A(V_f)}{\sum_{f=1}^{F} L_j(V_f) A(V_f)} \tag{10.3}$$

berechnet werden, wobei $L_j(V_f)$ der conditional likelihood der j-ten Person an dem Quadraturpunkt V_f entspricht.

10.2 Das Programm GGUM 2004

Das Programm GGUM 2004 – in der vorliegenden Version 1.1 – wurde entwickelt, um das soeben vorgestellte GGUM Modell von Roberts et al. (2000) empirisch zu schätzen. Es steht als Freeware unter http://www.psychology.gatech.edu/unfolding/ zum Herunterladen zur Verfügung.
Im Folgenden werden das Programm und seine Möglichkeiten anhand einer chronologisch aufgebauten Bedienungsanleitung beschrieben, d.h. es wird vorab kein allgemeiner Überblick gegeben, sondern gleich direkt mit der Programmkonfiguration begonnen. Dabei werden allerdings an Stellen, die als Besonderheit von Generalized Graded Unfolding Models bzw. des Programms GGUM 2004 an sich zu erwähnen sind, als zusätzliche Erläuterung eingefügt.

10.2.1 Spezifikation einer Schätzung

Zur Festlegung diverser Schätzparameter kann entweder eine schlicht gestaltete Windows Oberfläche verwendet werden, oder man liest ein zuvor erstelltes *Command-File* ein. In diesem werden alle benötigten Parameterwerte, einem speziellen Schema folgend, untereinander in einer Textdatei zusammengestellt und dann einmalig eingelesen (siehe dazu Abb. 10.3).

Die folgende Auflistung führt diesbezüglich jene Parameter an, die vor der Schätzung spezifiziert werden müssen. Die jeweilige Ordnungsnummer entspricht der Reihenfolge in der die jeweiligen Parameterwerte ggf. auch in einem *Command-File* angegeben werden müssen.

1. Auswahl des zu schätzenden Modells
   Das Programm GGUM 2004 ist in der Lage 8 verschiedene Modelle zu schätzen, wobei Nummer 8 (GGUM) das allgemeinste Modell darstellt. Die übrigen Modelle 1 bis 7 stellen jeweils in unterschiedlicher Weise restringierte Versionen des GGUM dar.

   Modell 1 *Constant Unit* Version des Generalized Graded Unfolding Model (GGUM)
   In der Constant Unit Version werden die Diskriminationsparameter α_i über alle Items

```
8              ESTIMATE PARAMETERS OF MODEL 8
N              CONSTRAINTS ARE NOT USED
N              DO NOT CHANGE THE SIGN OF INITIAL PARAMETER ESTIMATES
30             NUMBER OF QUADRATURE POINTS
K:\Doktoratsstudium\WS 09_10\Psychometric Methods\Data\cpdat.dat
(i4,1x,24i1)
24             NUMBER OF ITEMS
Y              IS NUMBER OF CATEGORIES CONSTANT?
6              NUMBER OF RESPONSE CATEGORIES
Y              DO YOU WANT TO RECODE THE DATA?
Y              IS RESPONSE CUTOFF CONSTANT?
0              RESPONSE CUTOFF
N              DISCARD ANY ITEMS
N              DISCARD ANY PEOPLE
N              SIGNS OF INITIAL LOCATION ESTIMATES NOT MANUALLY ASSIGNED
200            NUMBER OF OUTER CYCLES
10             NUMBER OF INNER CYCLES
30             NUMBER OF FISHER SCORING ITERATIONS FOR THRESHOLDS
30             NUMBER OF FISHER SCORING ITERATIONS FOR DELTAS & ALPHAS
0,001          CRITERION
Y              WANT TO PLOT
20             NUMBER OF PLOT GROUPS
2000           NUMBER OF THETA-DELTA PAIR GROUPS
Y              WANT FIT STATISTICS
3              NUMBER OF FIT GROUPS
Y              PRINT FIT FOR EVERY PERSON
2.576          ITEM T-VALUE CUTOFF
0.01           ITEM CHI-SQUARE PROBABILITY CUTOFF
3,291          PERSON T-VALUE CUTOFF
0,001          PERSON CHI-SQUARE PROBABILITY CUTOFF
2,576          PERSON LOCALIZED T-VALUE CUTOFF
```

Abb. 10.3 Beispiel eines *Command-Files*

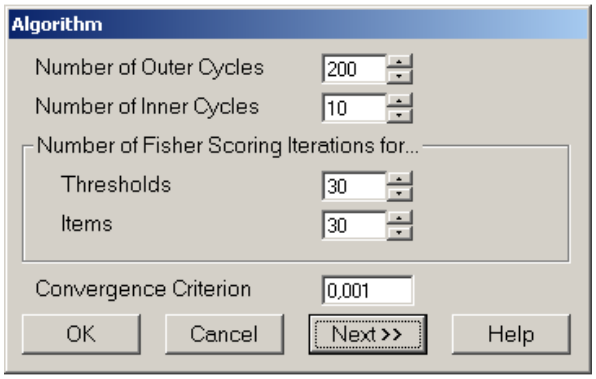


Abb. 10.4 Beispielansicht einer Windows-basierten Eingabemaske welche alternativ zum Einlesen eines *Command-Files* verwendet werden kann. Es lassen sich alle Parameterwerte auch über dieses Windows-basierte Benutzerinterface festlegen. Hier wird exemplarisch die Eingabe der Parameter 15 (Number of Outer Cycles) bis 19 (Convergence Criterion) dargestellt.

hinweg konstant auf 1 restringiert. Gleichzeitig werden die Abstände zwischen den einzelnen Thresholds konstant auf 2λ gehalten.

Modell 2 *Multiple Unit* Version des GGUM
Das Multiple Unit Modell erlaubt eine über die Items variierende Anzahl an Antwortkategorien. Gleichzeitig werden die Diskriminationsparameter über alle Items hinweg auf konstant 1 gesetzt und die Abstände der Thresholds eines Items i betragen $2\lambda_i$.

Modell 3 *Rating Scale* Version des GGUM
Wenn die Thresholds τ_{ik} über alle Items i gleich und alle α_i konstant 1 sind, spricht man vom Rating Scale Modell. Dieses ist auch unter dem Namen *Graded Unfolding Model* (GUM) von Roberts und Laughlin (1996) bekannt.

Modell 4 *Partial Credit* Version des GGUM
Wenn die Diskriminationsparameter α_i über alle Items i auf konstant 1 restringiert werden, erhält man die Partial Credit Version des GGUM.

Modell 5 *Generalized Constant Unit* Version des GGUM
Das Generalized Constant Unit Modell unterscheidet sich vom Constant Unit Modell nur insofern, als dass die Diskriminationsparameter α_i nicht mehr auf 1 restringiert sind, sondern über die Items variieren dürfen. Die Abstände zwischen den Thresholds sind weiterhin konstant 2λ.

Modell 6 *Generalized Multiple Unit* Version des GGUM
Die generalisierte Version des Multiple Unit Modells erlaubt die Variation des Diskriminationsparameters α_i über die alle Items. Die Abstände zwischen den Thresholds eines Items i betragen weiterhin $2\lambda_i$.

Modell 7 *Generalized Rating Scale* Version des GGUM
Das Generalized Rating Scale Modell lockert die Restriktion des Rating Scale Modells hinsichtlich der Diskriminationsparameter α_i. Diese dürfen in der verallgemeinerten Version über die Items i variieren.

Modell 8 *Generalized Graded Unfolding Model* (GGUM)
Das allgemeinste Modell, das mit GGUM 2004 zu schätzen ist, ist weder in den Diskriminationsparametern α_i, noch in den Thresholds τ_{ik} (bzw. in den Abständen der Thresholds λ_i) restringiert.

2. Sollen die Lambda- bzw. Threshold-Parameter restringiert werden?
Durch das Setzen dieses Parameters auf Y[=Yes] anstatt von N[=No], kann man die für die Modelle 2, 4, 6 und 8 die Bedingung hinzufügen, dass die Lambda-Werte der Items (obwohl sie weiterhin für jedes Item variieren dürfen) im Durchschnitt über alle Items konstant bleiben müssen.

3. Invertieren der ursprünglichen Parametervorzeichen?
Durch das Setzen von Y anstatt von N kann die Skala der Itemkategorien umgedreht werden; d.h. die Codierung von starker bis schwacher Zustimmung wird genau invertiert. Auf die Likelihood-Funktion hat dies keine Auswirkung.

4. Anzahl der Quadraturpunkte
Die MML und EAP Verfahren verwenden numerische Integrationsalgorithmen. Die Anzahl der Quadraturpunkte kann maximal 50 betragen und gibt sozusagen die „Auflösung“ an, die im Zuge des numerischen Integrationsverfahrens verwendet werden soll.

5. Datenquelldatei
Hier muss der vollständige Pfad der Datendatei in folgender Syntax angeben werden: *drive:\path\filename.ext*

6. FORTRAN input format zum Datenauslesen
   Die Datendatei muss grundsätzlich als plain text vorliegen. Damit GGUM jedoch weiß, wie die darin enthaltenen Zeichen zu interpretieren sind, d.h. wie die Codes für ID und Itemantworten vergeben wurden und wie sie im Textfile angeordnet sind, müssen derartige Merkmale der Datenstruktur mittels FORTRAN Code bekannt gegeben werden. Wie dieser zu verwenden ist, um ein vorliegendes Datenfile entsprechend einzulesen, kann der diesbezüglich sehr ausführlichen Hilfedatei des Programmes entnommen werden.

7. Anzahl der Items
   Bekanntgabe, für wie viele Items nun Daten eingelesen werden.

8. Ist die Anzahl der Antwortkategorien über alle Items konstant?
   GGUM könnte auch mit Items verschiedener Antwortkategorien umgehen. Es muss lediglich bekannt gegeben werden, ob die Anzahl der Antwortkategorien im konkreten Fall über alle Items konstant ist [Y] oder nicht [N].

9. Umkodierung der Daten notwendig?
   GGUM benötigt die Antwortkategorien im Format 0 bis C-1, wobei C die Anzahl der Anwortmöglichkeiten ist. Es könnte aber sein, dass die Daten tatsächlich im Format 1 bis C vorliegen. In diesem Fall wäre es erforderlich die Antworten zu recodieren (die Zahl 1 von jeder Antwortkategorie abzuziehen) um die Daten im Format 0 bis C-1 vorliegen zu haben.

10. Ist der Response Cutoff konstant?
    Der Response Cutoff gibt an, welche Antwortausprägung ein Individuum bei zumindest einem Item (aus dem Pool aller Items) jedenfalls angeben muss, um nicht von der Analyse exkludiert zu werden. Man könnte bei GGUM 2004 grundsätzlich für jedes Item einen separaten Cuttofflevel angeben. Durch setzen dieses constant cutoff Parameters auf Y [YES] wird jedoch ein konstanter Cutoff Parameter für alle Items verwendet.

11. Response Cutoff
    Mit diesem Parameter wird der Response Cutoff festgelegt. Wurde dieser als konstant über alle Items definiert, folgt nun eine Zahl. Wurde bei Frage 10 angegeben, dass der Response Cutoff nicht über alle Items konstant ist [N], dann folgen nun untereinander so viele Zahlen wie Items im Sample vorhanden sind, die die Response Cutoffs für jedes Item separat (jeweils im Format 0 bis C-1) festlegen.

12. Ausschließen einzelner Items
    Es können auch einzelne Items von der Analyse ausgeschlossen werden. Dazu setzt man zunächst ein Y und trägt in die darunter liegende Zeile die Anzahl der zu exkludierenden Items ein. Wiederum darunter folgt pro weiterer Zeile jeweils eine zu entfernende Itemnummer. Wird ein N gesetzt, so entfallen die Zeilen zum entfernen einzelner Items.

13. Ausschließen einzelner Personen
    In exakt gleicher Schreibweise lassen sich auch einzelne Respondenten von der Analyse ausschließen. Siehe dazu die Beschreibung unter Punkt 12.

14. Manuelles Festlegen der Vorzeichen aller ursprünglichen Lageschätzungen?
    Manchmal kann es wünschenswert sein, das Vorzeichen, d.h. die Richtung der Bewertung manuell umzukehren. Dies könnte zum Beispiel bei positiven und negativen Fragestellungen der Fall sein, die ein und dieselbe Variable abfragen sollen. Zu setzen ist hier ein [Y] oder ein [N]. Im ersten Fall folgt dann zusätzlich ein Block in dem in jeder Zeile (so viele wie Items) ein Plus oder Minus steht.

15. Anzahl der EM (Outer) cycles
    Spezifiziert die Anzahl der EM Iterationen, die durchlaufen werden müssen, bevor der Algorithmus abbricht. Hierbei wird ein Wert von (mind.) 200 empfohlen um die Konvergenz zu garantieren.

16. Anzahl der inner cycles
    Zunächst werden die Threshold-Parameter τ_{ik} geschätzt. Darauf folgend dann die Item-Parameter für Lage δ_i und Diskriminationsparameter α_i. Die einmalige Schätzung dieser drei Parameter bezeichnet einen inner cycle. Die Anzahl der inner cycles gibt an, wie viele Schätz-Iterationen durchlaufen werden müssen, bevor ein outer cycle abgeschlossen werden kann.

17. Anzahl der Fisher Scoring Iterationen für die Thresholds
    Die Schätzungen der Thresholds (τ_{ik}) werden mittels Fisher Scoring Verfahren durchgeführt. Die hier festzulegende Zahl gibt an, wie viele Fisher scoring iterationen innerhalb eines inner cycles durchgeführt werden. Von den Autoren der Software wird ein Wert von 30 vorgeschlagen.

18. Anzahl der Fisher Scoring Iterationen für die Items
    Hier wird, äquivalent zu den Threshold-Parametern (siehe 17), die Anzahl der Iterationsschritte für die Schätzung der Location-Parameter (δ_i) und Discrimination-Parameter(α_i) festgelegt.

19. Convergence Criterion
    Sowohl der Fisher Scoring Algorithmus, der inner und die outer Cycle Schätzungen brechen die Iterationen ab, sobald die Veränderung der Item-Parameter von einem zum nächsten Durchlauf absolut kleiner ist, als der hier spezifizierte Zahlenwert.

20. Grafische Plots erzeugen?
    GGUM 2004 bietet auch die Möglichkeit Plots zu erstellen. Um solche zu kreieren muss dies jedoch bereits vor der Schätzung bekannt gegeben werden. GGUM erwartet, falls Plots gewünscht sind [Y], noch zwei weitere Eingaben, die im *Command-File* einfach in die nächsten 2 Zeilen geschrieben werden. Zunächst muss die Anzahl der Plot-Gruppen festgelegt werden. Hierbei werden Personen mit ähnlichen θ-Werten gruppiert. Man kann die im Sample enthaltenen Personen in 2 bis 20 Gruppen aufteilen lassen. Ebenfalls werden für den Model-Fit-Plot die Unterschiede zwischen der Personen-Location und der Item-Location berechnet. Diese Unterschiede können auch nach Größe geclustert werden. Genauer gesagt kann man hier wieder angeben in wie viele homogene Gruppen aufgeteilt werden soll und hat die Wahl zwischen 2 und 2000.

21. Fit Statistiken berechnen?
    Diese Frage ist auch wieder mit Y [YES] und N [No] zu beantworten. Bei Y, ist wie schon bei Parameter 20 (Plots) auch hier die Anzahl der auf Basis der θ-Werte zu bildenden homogenen Gruppen zwischen 2 und 20 anzugeben. Des weiteren steht die Wahl zur Verfügung für jedes Subjekt Fit Statististiken auszugeben oder nur für einzelne, die gewisse, im Anlassfall zu definierende fit-Werte überschreiten.

10.2.2 Ein Beispieldatensatz

Der verwendete Beispieldatensatz ist auf der Homepage der GGUM Entwickler unter *http://www.psychology.gatech.edu/unfolding/data.html* zu finden.
Die verwendeten Itemaussagen stammen ursprünglich von Thurstone (1932) und wurden schließlich von Shaw und Wright (1967) neuerlich publiziert. Die dabei verwendeten Skalenwerte basierten dabei auf Zustimmungs- bzw. Ablehnungsergebnissen von 90 College Studierenden, die dazu aufgerufen waren jede ihnen vorgelegte Aussage auf einer 9-teiligen Skala von „very unfavorable“ bis „very

favorable“ zu bewerten.
Die im vorliegenden Beitrag verwendeten Umfrageergebnisse stammen von Roberts (1995). Dieser befragte 245 Studierende über ihren Zustimmungsgrad zu 24 Aussagen über die Sinnhaftigkeit der Einhebung von Geldstrafen. Dabei konnten die Antworten auf einer 6-teiligen Skala (1=“Strongly Disagree“, 2=“Disagree“, 3=“Slightly Disagree“, 4=“Slightly Agree“, 5=“Agree“ und 6=“Strongly Agree“) eingetragen werden. Tabelle 10.1 gibt einen Eindruck über das vorliegende Datenformat. (Anmerkung: Die Zeilennummer ist nicht Teil des Datensatzes sondern wurde nur zu illustrativen Zwecken hinzugefügt). Der verwendete Fragenkatalog ist diesem Kapitel als Appendix angeschlossen (siehe Seite 182).

Tabelle 10.1 Format der Rohdaten

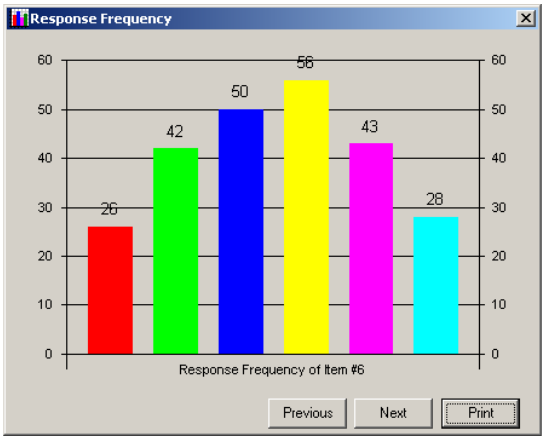
Zeile	ID (4-stellig), Lehrzeichen und Item-Antworten (24-stellig)
1	1058 436334333643233245446325
2	1059 514545154611222341154144
3	1060 355444352531325252252645
⋮	⋮
244	5107 216314111641112152154125
245	5108 313213132521222452352235

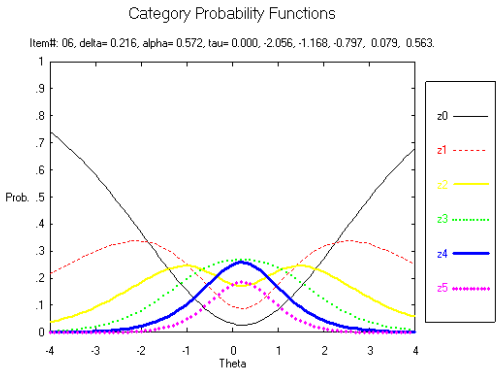
In Abbildung 10.5 sind einige grafische Ausgabemöglichkeiten von GGUM 2004 dargestellt. Neben der itemspezifischen Darstellung der Häufigkeitsverteilung der gegebenen Antworten lassen sich auch Item Characteristic Curves (ICC) und Item Information Funktionen ausgeben. Das Kernstück der Plots, insbesondere im Hinblick auf Entfaltungsmodelle, stellen aber sicherlich die Category Probability Functions dar, die durch die in Abschnitt 10.1.1 erläuterten Vorgangsweise, ihre spezifische Formen erhalten (vgl. Abbildung 10.2).
Es muss jedoch festgehalten werden, dass sich die Items des hier verwendeten Beispieldatensatzes (Meinungen zu Geldstrafen) zumindest insofern von den in Roberts et al. (2000) verwendeten (Meinungen zur Abtreibung) unterscheiden, als sie weder nach Einstellung (Lageparameter) geordnet, noch gleichmäßig über den möglichen Einstellungsraum verteilt sind. Dies liegt unter Anderem daran, dass einige Fragen in einer Weise gestellt wurden, dass sie kaum eine andere Möglichkeit als Ablehnung zuließen[1].

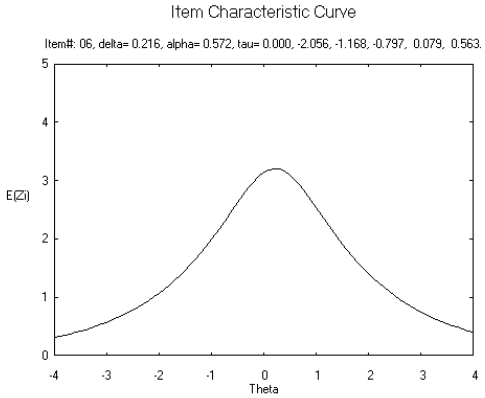
Neben den grafischen Ausgabemöglichkeiten (vgl. Abbildung 10.5) kann auch Textoutput ausgegeben werden. Neben einer *Execution Information*, die nochmals die gewählte Konfiguration zusammenfasst und den ordnungsgemäßen Ablauf bestetigt, stehen weiters folgende Tabellen zur Verfügung:

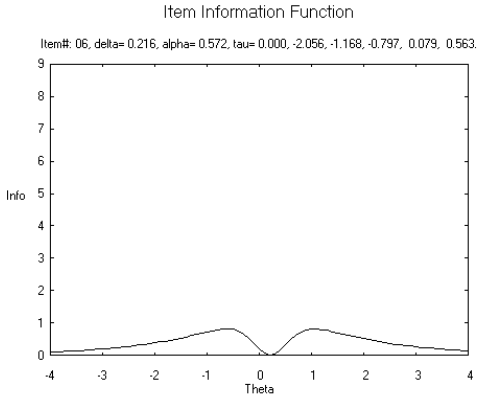
- Item-Parameter
  Man erhält eine Plaintext-Datei, die für jedes Item Diskriminationsparameter (α), Lageparameter (δ) sowie deren Standardabweichungen beinhaltet. Zusätzlich werden ebenfalls für jedes Item die Tresholds (τ) und deren Standardabweichungen angegeben.
- Personen-Parameter
  Hiermit erhält man die Schätzwerte für die Personen-Lageparameter und deren Standardabweichungen für alle Personen.
- Fit-Statistiken
  Für jedes Item und für jede Person können weiters einige Statistiken ausgegeben werden, wie gut das geschätzte Modell die empirischen Daten erklären kann.

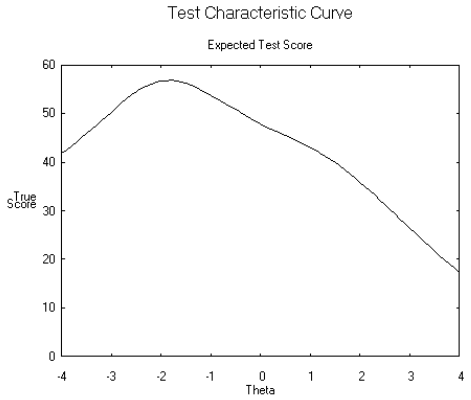
[1] Aussage 7 lautet z.B.: *Every criminal should be executed*.: hierbei dominiert Antwortkategorie „strongly disagree“ über alle abgebildeten Lagebereichsdifferenzen (-4 bis +4). Aussage 11: *I think the return of the whipping post would be more effective than capital punishment*. führt überhaupt zu nahezu waagrechten category probability functions mit einer klaren Dominanz von „strongly disagree“











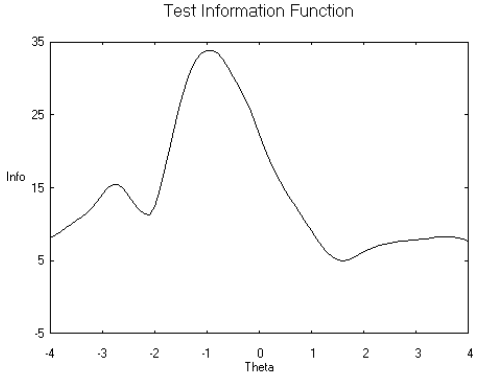


Abb. 10.5 Einige grafische Ausgabemöglichkeiten mit GGUM 2004, wie itemspezifischen Darstellung der Häufigkeitsverteilung der gegebenen Antworten, Plot der Wahrscheinlichkeitsfunktionen für Antwortkategorien eines Items, Item Characteristic Curves (ICC) sowie Item Information Funktionen ausgeben. Im Hinblick auf Entfaltungsmodelle gibt es auch Plots der Category Probability Functions.

10.2.3 Weitere (kommerzielle) Softwarepakete zur Schätzung von Unfolding Models

MUDFOLD - Multiple Unidimensional Unfolding

Das Programm MUDFOLD wurde von van Schuur und Molenaar (1982) und T. Wierstra (van Schuur und Wierstra, 1987) entwickelt. Es eignet sich zur Analyse von Einstellungs-, Präferenz- oder Entscheidungsdaten anhand des unfolding Modells von Coombs. Es können dabei verschieden generierte Daten wie complete oder partial rank orders, Likert-type rating scales oder dichotome Daten als Input verwendet werden. MUDFOLD sucht dann aus diesen Inputs eine Maximalzahl an Stimuli, die auf eine unfolding Dimension abgebildet werden können. Gleichzeitig wird auch begründet, warum einzelne, nicht repräsentierte Stimuli nicht abgebildet werden konnten.
MUDFOLD bietet eine Reihe von Goodness of Fit Tests an und hält auch die Möglichkeit bereit, ein ordered subset aller Stimuli (Items), die über alle Personen am besten zum unidimensionalen unfolding scale passen. In einer Analyse können maximal 36 Items und einige tausend Personen untersucht werden. GGUM kennt keine derartige Item-Beschränkung.

RUMMFOLDss und RUMMFOLDpp - Rasch Unidimensional Models for Measurement for Unfolding Response Models

RUMMFOLD wurde von David Andrich (Murdoch University) entwickelt. Es werden zwei Programmversionen unterschieden (ss und pp), die sich jeweils auf eine spezielle Art der Datengenerierung beziehen. Daten aus einem direct-response Single-Stimulus Design werden mit RUMMFOLDss und Daten aus pair-comparison bzw. pairwise preference Designs werden mit RUMMFOLDpp analysiert. Beide Programme schätzen letztlich die Item Location Parameter, jedoch sind aufgrund der unterschiedlichen Erhebungsverfahren (bei ss sind Items unabhängig von einander, während bei pp Daten aus Paarvergleichen resultieren) verschiedene Modelle notwendig. So wird für Paarvergleiche beispielsweise das *Rasch unfolding model for attitude and preference data* herangezogen, um die Item-Lageparamter zu bestimmen.
Sowohl RUMMFOLDss als auch RUMMFOLDpp verwenden ASCII Dateien als Input. Ausgegeben werden können neben den Lageparametern und Standardfehlern auch Teststatistiken zum Goodness of Fit. Es stehen aber auch einige grafische Outputs zu Personen und Itemparametern stehen zur Verfügung [2].

[2] Die Informationen zu MUDFOLD und RUMMFOLD stammen vom Softwarevertrieb Assessment Systems Corporation http://www.assess.com

Appendix A

Tabelle 10.2 Fragenkatalog von Roberts et al. (2000)

Nummer	Frage
1	Capital punishment may be wrong but it is the best preventative to crime.
2	Capital punishment is absolutely never justified.
3	I think capital punishment is necessary but I wish it were not.
4	Any person, man or woman, young or old, who commits murder, should pay with his own life.
5	Capital punishment cannot be regarded as a sane method of dealing with crime.
6	Capital punishment is wrong but is necessary in our imperfect civilization.
7	Every criminal should be executed.
8	Capital punishment has never been effective in in preventing crime.
9	I don't believe in capital punishment but I'm not sure it isn't necessary.
10	We must have capital punishment for some crimes.
11	I think the return of the whipping post would be more effective than capital punishment.
12	I do not believe in capital punishment under any circumstances.
13	Capital punishment is not necessary in modern civilization.
14	We can't call ourselves civilized as long as we have capital punishment.
15	Life imprisonment is more effective than capital punishment.
16	Execution of criminals is a disgrace to civilized society.
17	Capital punishment is just and necessary.
18	I do not believe in capital punishment but it is not practically advisable to abolish it.
19	Capital punishment is the most hideous practice of our time.
20	Capital punishment gives the criminal what he deserves.
21	The state cannot teach the sacredness of human life by destroying it.
22	It doesn't make any difference to me whether we have capital punishment or not.
23	Capital punishment is justified only for premeditated murder.
24	Capital punishment should be used more often than it is.

Literaturverzeichnis

Adams, R. J., Doig, B. A., und Rosier, M. (1991). *Science Learning in Victorian Schools*. Australian Council for Educational Research, Hawthorn, Victoria.

Adams, R. J., Wilson, M., und Wang, W.-C. (1997a). The multidimensional random coefficients multinomial logit model. *Applied Psychological Measurement*, 21(1):1–23.

Adams, R. J., Wilson, M. R., und Wu, M. L. (1997b). Multilevel item response modelling: An approach to errors in variables regression. *Journal of Educational and Behavioral Statistics*, 22:47–76.

Allan, S. J. und Hubbard, R. (1986). Regression Equations for the Latent Roots of Random Data Correlation Matrices with Unities on the Diagonal. *Multivariate Behavioral Research*, 21(3):393–396.

Andersen, E. B. (1970). Asymptotic properties of conditional maximum likelihood estimators. *Journal of the Royal Statistical Society, Series B*, 32:283–301.

Andersen, E. B. (1972). The numerical solution of a set of conditional estimation equations. *Journal of the Royal Statistical Society, Series B*, 34:42–54.

Andersen, E. B. (1973). A goodness of fit test for the Rasch model. *Psychometrika*, 38:123–140.

Andersen, E. B. (1983). A General Latent Structure Model for Contingency Table Data. In Wainer, H. und Messik, S., Hrsg., *Principals of Modern Psychological Measurement*, Seiten 117–138. Erlbaum, Hillsdale, NJ.

Andersen, E. B. (1995). Polytomous Rasch Models and their Estimation. In Fischer, G. und Molenaar, I., Hrsg., *Rasch models: Foundations, recent developments, and applications*, Seiten 271–292. Springer, New York.

Anderson, C., Li, Z., und Vermunt, J. (2007). Estimation of models in the Rasch family for polytomous items and multiple latent variables. *Journal of Statistical Software*, 20(6).

Andrich, D. (1978a). Application of a Psychometric Rating Model to Ordered Categories which are Scored with Successive Integers. *Applied Psychological Measurement*, 2(4):581–594.

Andrich, D. (1978b). A Rating Formulation for Ordered Response Categories. *Psychometrika*, 43(4):561–573.

Andrich, D. (1982). An Index of Person Separation in Latent Trait Theory, the Traditional KR-20 Index, and the Guttman Scale Response Pattern. *Education Research and Perspectives*, 9(1):95–104.

Andrich, D. (1988a). A General Form of Rasch's Extended Logistic Model for Partial Credit Scoring. *Applied Measurement in Education*, 1(4):363–378.

Andrich, D. (1988b). *Rasch Models for Measurement (Sage University paper series on quantitative applications in the social sciences)*. Sage, Newbury Park, CA.

Andrich, D. (1989). A Probabilistic IRT Model for Unfolding Preference Data. *Applied Psychological Measurement*, 13:193–216.

Andrich, D. (1995a). Further Remarks on Non-Dichotomization of Graded Responses. *Psychometrika*, 60(1):37–46.

Andrich, D. (1995b). Models for Measurement, Precision and the Non-Dichotomization of Graded Responses. *Psychometrika*, 60(1):7–26.

Andrich, D. (1996). A Hyperbolic Cosine Latent Trait Model for Unfolding Polytomous Responses: Reconciling Thurstone and Likert Methodologies. *British Journal of Mathematical and Statistical Psychology*, 49:347–365.

Andrich, D. und Luo, G. (2003). Conditional Pairwise Estimation in the Rasch Model for Ordered Response Categories using Principal Components. *Journal of Applied Measurement*, 4(3):205–221.

Andrich, D., Sheridan, B. S., und Luo, G. (2010). *Rumm 2030: Rasch Unidimensional Measurement Models*. RUMM Laboratory Perth, Western Australia.

Baker, F. B. und Kim, S.-H. (2004). *Item Response Theory: Parameter Estimation Techniques*. Marcel Dekker, Reading, NY, 2nd edition.

Bartholomew, D., Steele, F., Moustaki, I., und Galbraith, J. (2002). *The Analysis and Interpretation of Multivariate Data for Social Scientists*. Chapman & Hall, London.

Bédard, M., Molloy, D. W., Squire, L., Dubois, S., Lever, J. A., und O'Donnell, M. (2001). The Zarit Burden Interview: A New Short Version and Screening Version. *The Gerontologist*, 41(5):652–657.

Bejar, I. (1983). *Achievment testing*. Beverly Hills: Sage.

Ben-Simon, A., Budescu, D. V., und Nevo, B. (1997). A Comparative Study of Measure of Partial Knowledge in Multiple-Coice Tests. *Applied Psychological Measurement*, 21(1):65–88.

Bereby-Meyer, Y., Meyer, J., und Flascher, O. M. (2002). Prospect Theory Analysis of Guessing in Multiple Choice Tests. *Journal of Behavioral Decision Making*, 15(4):313–327.

Bühner, M. (2006). *Einführung in die Test- und Fragebogenkonstruktion*. Pearson Studium.

Birnbaum, A. (1968). Some Latent Trait Models and Their Use in Inferring an Examinee's Ability. In Lord, F. M. und Novick, M. R., Hrsg., *Statistical Theories of Mental Test Scores*, Seiten 395–479. Addison-Wesley, Reading, MA.

Bock, R. D. (1972). Estimating item parameters and latent ability when responses are scored in two or more nominal categories. *Psychometrika*, 37(1):29–51.

Bock, R. D. und Aitkin, M. (1981). Marginal Maximum Likelihood Estimation of Item Parameters: Application of an EM Algorithm. *Psychometrika*, 46:443–445.

Bond, T. G. und Fox, C. M. (2007). *Applying the Rasch model*. Lawrence Erlbaum Associates, Mahwah, NJ, second edition.

Borsboom, D. (2006). The attack of the psychometricians. *Psychometrika*, 71:425–440.

Brook, L., Taylor, B., und Prior, G. (1991). *British Social Attitudes, 1990, Survey*. SCPR, London.

de Boeck, P. und Wilson, M. (2004). *Explanatory item response models: A generalized linear and nonlinear approach*. Springer, New York.

de Leeuw, J. (2005). Multidimensional Unfolding. In *Encyclopedia of Statistics in Behavioral Science*. Wiley.

de Leeuw, J. und Verhelst, N. (1986). Maximum likelihood estimation in generalized Rasch models. *Journal of educational statistics*, 11:183–196.

Doran, H., Bates, D., Bliese, P., und Dowling, M. (2007). Estimating the Multilevel Rasch Model: With the `lme4` Package. *Journal of Statistical Software*, 20(2):1–18.

Eckes, T. (2003). Qualitätssicherung beim TestDaF: Konzepte, Methoden, Ergebnisse. *Fremdsprachen and Hochschule*, 69:43–68.

Fischer, G. (1983). Logistic linear trait models with linear constraints. *Psychometrika*, 48:3–26.

Fischer, G. H. (1973). The linear logistic test model as an instrument in educational research. *Acta Psychologica*, 37:359–374.

Fischer, G. H. (1974). *Einführung in die Theorie psychologischer Tests*. Huber, Bern.

Fischer, G. H. (1977). Linear Logistic Trait Models: Theory and Application. In Spada, H. und Kempf, W. F., Hrsg., *Structural Models of Thinking and Learning*, Seiten 203–225. Huber, Bern.

Fischer, G. H. (1981). On the existence and uniqueness of maximum-likelihood estimates in the Rasch model. *Psychometrika*, 46:59–77.

Fischer, G. H. (1987). Applying the principles of specific objectivity and of generalizability to the measurement of change. *Psychometrika*, 52:565–587.

Fischer, G. H. (1988). Spezifische Objektvität: Eine wissenschaftstheoretische Grundlage des Rasch-Modells. In Kubinger, K., Hrsg., *Moderne Testtheorie*, Seiten 87–111. Beltz, Weinheim.

Fischer, G. H. (1995a). Derivations of the Rasch Model. In Fischer, G. und Molenaar, I., Hrsg., *Rasch Models: Foundations, Recent Developments, and Applications*, Seiten 15–38. Springer, New York.

Fischer, G. H. (1995b). Linear Logistic Models for Change. In Fischer, G. und Molenaar, I., Hrsg., *Rasch Models: Foundations, Recent Developments, and Applications*, Seiten 157–180. Springer, New York.

Fischer, G. H. und Molenaar, I. W. (1995). *Rasch models: Foundations, recent developments and applications*. New York: Springer.

Fischer, G. H. und Parzer, P. (1991). An extension of the rating scale model with an application to the measurement of change. *Psychometrika*, 56:637–651.

Fischer, G. H. und Ponocny, I. (1994). An extension of the partial credit model with an application to the measurement of change. *Psychometrika*, 59:177–192.

Fischer, G. H. und Ponocny-Seliger, E. (1998). *Structural Rasch Modeling: Handbook of the Usage of LPCM-WIN 1.0*. ProGAMMA, Groningen.

Fischer, G. H. und Scheiblechner, H. H. (1970). Algorithmen und Programme für das probabilistische Testmodell von Rasch. *Psychologische Beiträge*, 12:23–51.

Fox, J.-P. (2007). Multilevel IRT Modeling in Practice with the Package `mlirt`. *Journal of Statistical Software*, 20(5):1–16.

Frary, R. B. (1989). The Effect of Misinformation, Partial Information, and Guessing on Expected Multiple-Choice Test Item Scores. *Applied Psychological Measurement*, 4(1):79–90.

Glas, C. A. W. (1992). A Rasch Model with a Multivariate Distribution of Ability. In Wilson, M., Hrsg., *Objective Measurement: Theory into Practice, Volume 1*, Seiten 236–258. Ablex, Norwood, NJ.

Glas, C. A. W. und Verhelst, N. (1989). Extensions of the partial credit model. *Psychometrika*, 54:635–659.

Glas, C. A. W. und Verhelst, N. (1995a). Testing the Rasch model. In Fischer, G. und Molenaar, I., Hrsg., *Rasch Models: Foundations, Recent Developments, and Applications*, Seiten 69–96. Springer, New York.

Glas, C. A. W. und Verhelst, N. (1995b). Tests of Fit for Polytomous Rasch Models. In Fischer, G. und Molenaar, I., Hrsg., *Rasch Models: Foundations, Recent Developments, and Applications*, Seiten 325–352. Springer, New York.

Gustafsson, J. (1980). Testing and obtaining fit of data to the Rasch model. *British Journal of Mathematical and Statistical Psychology*, 33:205–233.

Haberman, S. J. (1977). Maximum likelihood estimates in exponential response models. *The Annals of Statistics*, 5:815–841.

Hoijtink, H. (1995). Linear and Repeated Measures Models for the Person Parameter. In Fischer, G. und Molenaar, I., Hrsg., *Rasch Models: Foundations, Recent Developments, and Applications*, Seiten 203–214. Springer, New York.

Hoijtink, H. und Boomsma, A. (1995). On person parameter estimation in the dichotomous Rasch model. In Fischer, G. und Molenaar, I., Hrsg., *Rasch Models: Foundations, Recent Developments, and Applications*, Seiten 53–68. Springer, New York.

Humphry, S. (2005). *Maintaining a Common Arbitrary Unit in Social Measurement*. PhD thesis, Murdoch University, Australia.

Jannarone, R. J. (1986). Conjunctive item response theory model kernels. *Psychometrika*, 51:357–373.

Katz, S., Ford, A. B., Moskowitz, R. W., Jackson, B. A., und Jaffe, M. W. (1963). Studies of Illness in the Aged. The Index of ADL: A Standardized Measure of Biological and Psychological Function. *JAMA*, 185(12):914–919.

Klassen, D. und O'Connor, W. A. (1989). Assessing the risk of violence in released mental patients: A cross-validation study. *Psychological Assessment: A Journal of Consulting and Clinical Psychology*, 1(2):75–81.

Kubinger, K. D. (1989). Aktueller Stand und kritische Würdigung der Probabilistischen Testtheorie. In Kubinger, K., Hrsg., *Moderne Testtheorie: Ein Abriss samt neuesten Beiträgen*, Seiten 19–83. Beltz, Weinheim.

Kubinger, K. D. (2005). Psychological test calibration using the Rasch model: Some critical suggestions on traditional approaches. *International Journal of Testing*, 5:377–394.

Linacre, J. (1994a). Constructing measurement with a many-facet Rasch model. In Wilson, M., Hrsg., *Objective Measurement: Theory in Practice*. Ablex, Newark, NJ.

Linacre, J. M. (1989). *Many-facet Rasch measurement*. MESA Press, Chicago.

Linacre, J. M. (1991). Structured Rating Scales. In *ERIC TM 016615*, Chicago, Illinois. International Objective Measurement Workshops.

Linacre, J. M. (1993). Generalizability Theory and Many-facet Rasch Measurement. *Paper presented at the Annual Meeting of the American Educational Research Association*.

Linacre, J. M. (1994b). *Many-facet Rasch measurement*. MESA Press, Chicago.

Linacre, J. M. (1997). *Judging plans and facets*. University of Chicago, MESA Psychometric Laboratory, Chicago.

Linacre, J. M. (2004a). Estimation Methods for Rasch Measures. In Smith Jr., E. V. und Smith, R. M., Hrsg., *Introduction to Rasch Measurement*, Seiten 25–48. JAM Press, Maple Grove, MN.

Linacre, J. M. (2004b). From Microscale to Winsteps: 20 years of Rasch Software. *Rasch Measurement Transactions*, 17(4):958.

Linacre, J. M. und Wright, B. D. (1994). Chi-Square Fit Statistics. *Rasch Measurement Transactions*, 8(2):350.

Liou, M. (1994). More on the computation of higher-order derivatives of the elementary symmetric functions in the Rasch model. *Applied Psychological Measurement*, 18:53–62.

Lunz, M., Wright, B. D., und Linacre, J. M. (1990). Measuring the impact of judge severity on examination scores. *Applied Measurement in Education*, 3:331–345.

Mair, P. (2006). Simulation Studies for Goodness-of-Fit Statistics in Item Response Theory. Master's thesis, Department of Psychology, University of Vienna.

Mair, P. und Hatzinger, R. (2007a). Extended Rasch Modeling: The eRm package for the application of IRT models in R. *Journal of Statistical Software*, 20(9):1–20.

Mair, P. und Hatzinger, R. (2007b). CML based estimation of extended Rasch models with the eRm package in R. *Psychology Science*, 49:26–43.

Mair, P. und Treiblmaier, H. (2008). Partial Credit Models for Scale Construction in Hedonic Information Systems. Technical Report 62, Department of Statistics and Mathematics Wirtschaftsuniversität Wien.

Masters, G. N. (1982). A Rasch Model for Partial Credit Scoring. *Psychometrika*, 47(2):149–174.

Mislevy, R. J. (1985). Estimation of latent group effects. *Journal of the American Statistical Association*, 80:993–997.

Mislevy, R. J. und Bock, R. D. (1990). *BILOG 3: Item Analysis and Test Scoring with Binary Logistic Models*. Scientific Software International, Inc., Chicago.

Molenaar, I. (1995). Estimation of Item Parameters. In Fischer, G. und Molenaar, I., Hrsg., *Rasch models: Foundations, recent developements, and applications*, Seiten 39–51. Springer, New York.

Muraki, E. (1992). A generalized partial credit model: Application of an EM algorithm. *Applied Psychological Measurement*, 16:159–176.

Pepels, W. (2007). *Market Intelligence – Moderne Marktforschung für Praktiker: Auswahlverfahren – Datenerhebung – Datenauswertung – Praxisanwendungen – Marktprognose*. PUBLICIS KommunikationsAgentur.

Pfanzagl, J. (1994). On Item Parameter Estimation in Certain Latent Trait Models. In Fischer, G. und Laming, D., Hrsg., *Contributions to Mathematical Psychology, Psychometrics, and Methodology*, Seiten 249–263. Springer, New York.

Pinheiro, J. und Bates, D. (1995). Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model. *Journal of Computational and Graphical Statistics*, 4:12–35.

Poinstingl, H., Mair, P., und Hatzinger, R. (2007). *Manual zum Softwarepackage eRm: Anwendung des Rasch-Modells*. Pabst Science Publishers, Lengerich.

Ponocny, I. (2001). Nonparametric goodness-of-fit tests for the Rasch model. *Psychometrika*, 66:437–460.

R Development Core Team (2007). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Rasch, G. (1960). *Probabilistic Models for some Intelligence and Attainment Tests*. Danish Institute for Educational Research, Copenhagen.

Rasch, G. (1961). On General Laws and the Meaning of Measurement in Psychology. In *Proceedings of the IV. Berkeley Symposium on Mathematical Statistics and Probability, Vol. IV*, Seiten 321–333. University of California Press, Berkeley.

Rasch, G. (1977). On Specific Objectivity: an Attempt at Formalizing the Request for Generality and Validity of Scientific Statements. *Danish Yearbook of Philosophy*, 14:58–93.

Rasch, G. (1980). *Probabilistic Models for Some Intelligence and Attainment Tests*. The University of Chicago Press, Chicago. expanded edition with foreword and afterword by B. D. Wright.

Reid, F. J. (1976). Scoring Multiple-Choice Exams. The Journal of Economic Education. *The Journal of Economic Education*, 8(1):55–59.

Reimers, H. (2005). *Winmira Online Manual*. Christian Albrecht Universität zu Kiel - Institut für Pädagogik.

Rizopoulos, D. (2006). ltm: An R package for latent variable modeling and item response theory analyses. *Journal of Statistical Software*, 17(5):1–25.

Rizopoulos, D. und Moustaki, I. (2008). Generalized Latent Variable Models with Non-Linear Effects. *British Journal of Mathematical and Statistical Psychology*, 61:415–438.

Roberts, J. S. (1995). *Item response theory approaches to attitude measurement*. PhD thesis, University of South Carolina, Columbia.

Roberts, J. S., Donoghue, J. R., und Laughlin, J. E. (2000). A General Item Response Theory Model for Unfolding Unidimensional Polytomous Responses. *Applied Psychological Measurement*, 24:3–32.

Roberts, J. S. und Laughlin, J. E. (1996). A unidimensional item response model for unfolding responses from a graded disagree-agree response scale. *Applied Psychological Measurement*, 20:231–255.

Rost, J. (1999). Was ist aus dem Rasch-Modell geworden? *Psychologische Rundschau*, 50:140–156.

Rost, J. (2000). The Growing Family of Rasch Models. In Boomsma, A., van Duijn, M., und Snijders, T., Hrsg., *Essays on item response theory*, Seiten 25–42. Springer, New York.

Rost, J. (2004). *Lehrbuch Testtheorie – Testkonstruktion*. Verlag Hans Huber, Bern.

Rost, J. und von Davier, M. (1995). Polytomous Mixed Rasch Models. In Fischer, G. und Molenaar, I., Hrsg., *Rasch Models: Foundations, Recent Developments, and Applications*, Seiten 371–382. Springer, New York.

Samejima, F. (1969). Estimation of Latent Ability using a Response Pattern of Graded Scores. *Psychometrika Monograph Supplement*, 34.

Scheiblechner, H. (1972). Das Lernen und Lösen komplexer Denkaufgaben. *Zeitschrift für Experimentelle und Angewandte Psychologie*, 3:456–506.

Shaw, M. E. und Wright, J. M. (1967). *Scales for the measurement of attitudes*. McGraw-Hill, NY.

Smith, E. V. (2002). Understanding Rasch Measurement: Detecting and Evaluating the Impact of Multidimensionality using Item Fit Statistics and Principal Component Analysis of Residuals. *Journal of Applied Measurement*, 3(2):205–230.

Stelzl, I. (1993). Testtheoretische Module. In Tent, L; Stelzl, I., Hrsg., *Pädagogisch-psychologische Diagnostik*, Seiten 39–201. Göttingen: Hogrefe.

Suárez-Falcón, J. C. und Glas, C. A. W. (2003). Evaluation of global testing procedures for item fit to the Rasch model. *British Journal of Mathematical and Statistical Society*, 56:127–143.

Thissen, D. (1991). *MULTILOG: multiple category item analysis and test scoring using item response theory*. Scientific Software International, Inc., Chicago.

Thissen, D. und Steinberg, L. (1984). A response model for multiple choice items. *Psychometrika*, 49(4):501–519.

Thurstone, L. L. (1932). *Motion pictures and the attitudes of children*. University of Chicago Press, Chicago.

Trukeschitz, B., Mühlmann, R., Schneider, U., Ponocny, I., und Österle, A. (2009). Arbeitsplätze und Tätigkeitsmerkmale berufstätiger pflegender Angehöriger - Befunde aus der Wiener Studie zur informellen Pflege und Betreuung älterer Menschen 2008 (VIC2008). Technical Report 2, Forschungsinstitut für Altersökonomie, Wien.

van der Linden, W. und Hambleton, R. (1997). *Handbook of Modern Item Response Theory*. Springer, New York.

van Schuur, W. und Molenaar, I. (1982). MUDFOLD: Multiple stochastic unidimensional unfolding. In Caussinus, H., Ettinger, P., und Thomassone, R., Hrsg., *COMPSTAT 1982 Part I: Proceedings in computational statistics*, Seiten 419–426. Physica-Verlag, Vienna. Appeared also in: C.P. Middendorp, B. Niemoller, & W.E. Saris.(eds), Sociometric research 1982. Amsterdam: Sociometric Society, 1982, 222-228.

van Schuur, W. und Wierstra, T. (1987). *MUDFOLD manual*. Groningen: I.E.C. Gamma.

Verhelst, N. und Glas, C. A. W. (1995). The one parameter logistic test model. In Fischer, G. und Molenaar, I., Hrsg., *Rasch Models: Foundations, Recent Developments, and Applications*, Seiten 215–238. Springer, New York.

Verhelst, N., Glas, C. A. W., und Vries, H. H. D. (1997). A Steps model to analyze partial credit. In van der Linden, W. und Hambleton, R., Hrsg., *Handbook of Modern Item Response Theory*, Seiten 123–138. Springer, New York.

Verhelst, N., Hatzinger, R., und Mair, P. (2007). The Rasch sampler. *Journal of Statistical Software*, 20(4):1–14.

Volodin, N. A. und Adams, R. J. (1995). Identifying and estimating a D-dimensional item response model. *Paper presented at the International Objective Measurement Workshop, University of California, Berkeley, California*.

von Davier, M. (1997). *Methoden zur Prüfung probabilistischer Testmodelle*. IPN Schriftenreihe Band 157, Universität Kiel.

von Davier, M. (1998). *WINMIRA: A Windows program for mixed Rasch models*. IPN, Kiel.

von Davier, M. (2001). *Winmira 32 Manual*.

von Davier, M. (2006). Book Review: Introduction to Rasch Measurement. *Applied Psychological Measurement*, 30(5):443–446.

Wang, P. Y., Hrsg. (1995). *Modern test theory*. Psychological Publishing Co., Taipei.

Wang, X., Bradlow, E., und Wainer, H. (2005). *User's guide for SCORIGHT (Version 3.0): A computer program for scoring tests built of testlets including a module for covariate analysis*. Educational Testing Services, Princeton, NJ.

Warm, T. A. (1989). Weighted likelihood estimation of ability in item response theory. *Psychometrika*, 54:427–450.

Watkins, M. (2000). *Monte Carlo PCA for Parallel Analysis*. Ed & Psych Associates, State College, PA.

Wewers, M. E. und Lowe, N. K. (1990). A critical review of visual analogue scales in the measurement of clinical phenomena. *Research in Nursing and Health*, 13:227–236.

Wilson, M. (1992). The ordered partition model: An extension of the partial credit model. *Applied Psychological Measurement*, 16:309–325.

Winsteps (n.d.). Winsteps & Facets Comparison.

Wright, B. D. (1998). *Introduction to the Rasch model*. MESA Press, Chicago.

Wright, B. D. und Masters, G. (1990). Computation of OUTFIT and INFIT Statistics. *Rasch Measurement Transactions*, 3(4):84–85.

Wright, B. D. und Masters, G. (2002). Number of person or item strata. *Rasch Measurement Transactions*, 16(3):888.

Wright, B. D. und Masters, G. N. (1982). *Rating Scale Analysis: Rasch Measurement*. MESA Press, Chicago.

Wright, B. D. und Panchapakesan, N. (1969). A procedure for sample-free item analysis. *Educational and Psychological measurement*, 29:23–48.

Wright, B. D. und Stone, M. (1979). *Best Test Design: Rasch Measurement*. MESA Press, Chicago.

Wright, B. D. und Stone, M. H. (1999). *Measurement Essentials*. Wide Range Inc., Wilmington, Delaware.

Wu, M. L., Adams, R. J., Wilson, M., und Haldane, S. (2007). *ConQuest Version 2.0. [Computer software]*. Assessment Systems Corporation, St. Paul, MN.

Zimowski, M. F., Muraki, E., Mislevy, R. J., und Bock, R. D. (1996). *BILOG-MG: Multiple-group IRT analysis and test maintenance for binary items*. Scientific Software International, Inc., Chicago.

Zwindermann, A. H. (1995). Pairwise Parameter Estimation in Rasch Models. *Applied Psychological Measurement*, 19:369–375.